INCREASING THE EFFECTIVENESS OF EDUCATIONAL TECHNOLOGIES WITH THE USE OF MACHINE LEARNING METHODS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Mariheida Córdova Sánchez

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2017

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Luo Si, Co-Chair

School of Science

Dr. Hubert E. Dunsmore, Co-Chair

School of Science

Dr. Bharat Bhargava

School of Science

Dr. Dan Goldwasser

School of Science

Approved by:

Dr. William J. Gorman

Head of the Departmental Graduate Program

To my siblings, Natalia and Javier Andrés,
For always believing in me and for being my inspirations.

To my parents, Anneliesse and Javier,
For teaching me the value of hard work and for giving me everything.

ACKNOWLEDGMENTS

I would like to first thank my advisors, Dr. Luo Si and Dr. Hubert E. Dunsmore, for their support and mentoring. I am deeply grateful to them for all they have taught me throughout my time at Purdue. Their time, patience, and advice have been incredibly valuable to me.

I would like to thank the members of my Ph.D. advisory committee, Dr. Dan Goldwasser and Dr. Bharat Bhargava, for their valuable feedback and for discussing research ideas with me.

Thanks to my lab mates from whom I learned a great deal: Suleyman Cetintas, Ahmet Bugdayci, Yi Fang, Dan Zhang, Bin Shen, Dzung Hong, Ning Zhang, and Zhiwei Zhang. Thanks for the conversations and for making the lab a fun place to be at. Thanks to my coworkers and friends at ITaP for the great research ideas and discussions.

A very special thanks goes to Rohit, who had the impossible job of being both an advisor and a husband and who, although it did not always seem like it to me at the time, did a fantastic job.

I would like to thank all the staff members of the Department of Computer Science, especially Dr. William J. Gorman, Renate Mallus, Sandra Freeman, and Tammy Muthig, for all their help in making the process easier so I could focus on my work.

My most heartfelt thanks goes to my family. To my parents, thanks for always believing in me and for offering me so much support and encouragement through the years. To my sister, Natalia, thanks for being my greatest role model and for encouraging me even when you sometimes needed it yourself. To my brother, Javier Andrés, thanks for teaching me to fight for what you believe in. Thanks to my Indian family, who has given me a great amount of support and encouragement.

To the many wonderful friends I made during my time at Purdue, Zuli, Luis, Pedro, Kelly, Sindhura, Pawan, Soumyadip, Joo Young, Ahmet, Sebastián, Paulina, thanks for making this period of my life an enjoyable one. Thanks to the Puerto Rican Student Association and to the Latino Cultural Center for bringing the warmth of the Caribbean to the Midwest. And to the friends I carried with me from before my graduate studies, Yariza, Yelenna, and Barbara, thanks for always being there for me.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

ABBREVIATIONS

AUC	Area Under the Curve
BOW	Bag of Words
CMS	Course Management System
ITS	Intelligent Tutoring System
KC	Knowledge Component
kNN	k-Nearest Neighbors
LDA	Latent Dirichlet Allocation
ROC	Receiver Operating Characteristic
SVM	Support Vector Machines
TF	Term Frequency
TF-IDF	Term Frequency —Inverse Document Frequency

ABSTRACT

Córdova Sánchez, Mariheida Ph.D., Purdue University, May 2017. Increasing the Effectiveness of Educational Technologies with the Use of Machine Learning Methods. Major Professors: Luo Si and Hubert E. Dunsmore.

There is a vast amount of educational technologies that assist students and instructors in their academic experiences. Technologies have been developed in the recent past to help students carry out discussions inside and outside of the classroom, allowing students to voice their thoughts and get their questions answered. Other technologies known as intelligent tutoring systems adapt to students' abilities and provide personalized learning experiences. These technologies, while highly convenient, most of the time require a great amount of domain expert work to set up and maintain in order to be effective. In order to be most effective, educational technologies should *i*) prioritize relevant content in order to reduce distractions in classrooms; *ii*) systematically assist in labeling important contextual data; and *iii*) provide mechanisms to give feedback to the learners to improve their learning experiences and improve graduation rates.

We propose modeling classroom content data, in the form of discussions being held by students, intelligent tutoring systems' questions, as well as students' institutional data, such as their demographics and academic records, in order to improve student success. In this dissertation, we discuss the use of machine learning techniques to model and analyze this data gathered from educational technologies to feed back to the students and instructors, with the ultimate goal of improving the students' academic experience and learning process.

This dissertation studies whether we can effectively use machine learning methods to increase the effectiveness of educational technologies. Specifically, we study how

machine learning methods can be used to *i*) promote the most relevant and diverse questions or discussions held during classes in order to avoid distractions caused by irrelevant content and reduce the amount of time required to maintain; *ii*) assist in labeling content for adaptive learning or intelligent tutoring systems in order to reduce the time required to set up and allow for a more effective functioning of the tutoring algorithms; and *iii*) identify students who are at risk of not graduating from their current academic program, so as to be able to provide early warnings to students and increase retention and student success.

1 INTRODUCTION

Educational technologies provide mechanisms that go far beyond what is possible in a traditional classroom. These technologies are extremely valuable when it comes to student success. Some educational technologies allow students to carry out in-class discussions, where they could raise questions or even answer other students' questions during the class. Others provide students with personalized learning, allowing them to receive one-on-one teaching instructions which adapt to their knowledge acquisition. Others identify students who are struggling and at risk of not completing their academic programs, in order to provide these students with early warnings along with the resources they need to continue and be successful.

While educational technologies assist teachers and educational institutes in delivering the best learning experiences to students and assist students in getting the most out of their learning exercises, there are several reasons for instructors to be hesitant in the adoption of such technology. These reasons have been thoroughly studied in the past [1–3] and most studies agree that the main reasons for not adopting educational technology are: i) technology causing distracting for students, ii) the amount of human work that comes with this adoption being unreasonable, and iii) not seeing the direct benefit of adopting the technology. This chapter introduces these core problems as well as novel techniques to address them.

In this dissertation, we study some challenges present in educational technologies and apply machine learning methods to increase their efficiency taking into account the points stated above. Our main contributions are:

- Reduce distractions cause by online discussions held in class and reduce the time required to manage them by promoting the most relevant and diverse questions;

- Improve the functioning of intelligent tutoring systems and reduce the time required to set them up by assisting in tagging content;
- Identify students who are at risk of not graduating or at risk of changing majors.

In the following sections, we describe these challenges further and outline the rest of this dissertation.

1.1 Reduce Distractions in Online Course Discussions

Some applications allow students to carry out in-class discussions by providing a back-channel communication platform. These applications are helpful in allowing students to voice their questions or thoughts in settings where they would not otherwise, perhaps because they are afraid to raise their hands in a large classroom, or because they think their question might be inadequate. In some cases, these technologies even encourage students to ask questions by allowing them to post them anonymously. Moreover, instructors do not always want the students to ask questions directly to the instructor as they arise, as is many times the case for very large classes, since this might happen very frequently and would be disruptive to the flow of the class.

These back-channel communication tools provide just-in-time feedback to students by showing answers to their questions, at the expense of having the instructor or teaching assistant review and address the questions and comments. The increased review material and time required to address the questions can be a burden to the instructor or teaching assistant who only have limited time to respond during the lecture. Another problem introduced by online course discussions is that many times students post irrelevant material. Students might, for instance, post humorous comments or ask questions about the logistics of the course. This is not only distracting but could bury the actual relevant content, *i.e.*, the questions that need to be addressed during the lecture.

We propose solving both challenges described above by finding the most relevant and diverse content to the course material. That is, we want to promote the most relevant quesstions, while also covering different aspects or topics in the top ranked questions. This would save the instructor valuable time of reviewing the content to find the most important questions to address. It would also reduce distractions caused by irrelevant posts and by repeated quesitons. We look at in-class discussions collected by a web application developed at Purdue University, called Hotseat [4]. In this application, students are able to post their questions or comments, reply to other posts (answering other students' questions), post questions anonymously, and vote for other posts (presumably because they would also like this question to be answered). We use a combination of features extracted from the posts, such as the length of the post, the number of votes and replies, and whether they were posted anonymously. We also use topic modeling to find latent topics in both the lecture material and the posts, as well as features extracted from the lecture slides presented in class. We then train a model which learns what are the most relevant questions asked during the class. Another approach we develop consists in the use of submodularity to diversify the topics present in the questions shown at the top of the feed. This way we cover all relevant topics of the lecture, while giving less priority to redundant questions or comments. Our empirical analyses show that with both approaches we are able to identify the most relevant and diverse questions asked in a class with high accuracy. This study is presented in Chapter 2.

1.2 Assist in Tagging Content in Intelligent Tutoring Systems

Intelligent tutoring systems, or ITS, are tutoring systems that mimic a human teacher. They adapt future problems assigned to students based on the skills of the students. For example, if the student was not able to answer a question correctly, the intelligent tutor might offer prompts to help the student solve the problem, or provide an easier problem to solve, or provide a problem that involves the same skills

in a different context. These tools have been shown to be highly effective. In some cases they are even as effective as having a one-on-one human tutor [5], without the resources that a human tutor for each student would require. However, these tutors usually require a great amount of human expert work in order to be effective. For instance, not only does the expert need to input all the problems for the students to solve, but also needs to tag each problem with the associated skills or knowledge components required to solve them. Knowledge components (KC), are the skills or mental processes involved in the solving of a problem [6, 7]. In order for the ITS algorithms to assess knowledge mastery and decide which is the appropriate problem to assign next to the student, the problems need to be properly tagged with the KCs required to correctly solve the problem. Moreover, tagging educational content with KCs or skills is key to providing usable reports to teachers. With many systems using fine-grained KC models that range from dozens to hundreds of KCs to select from, the task of tagging new content with KCs can be a laborious and time consuming one. This can often result in content being left untagged.

We propose a system to assist content developers with the task of assigning KCs by suggesting KCs for new problems based on the text of the problems and their similarity to other expert-labeled content already on the system. Two approaches are explored for the suggestion engine. The first uses text mining and support vector machines. The second uses language modeling and a k-nearest neighbors algorithm. Our empirical results support the effectiveness of our proposed models. This work is presented in Chapter 3.

1.3 Identify Students Who Are Struggling

Early intervention systems can be very powerful tools for teaching institutions to use in identifying students who are at risk of not completing their academic program. These systems are designed to be able to identify these students early so as to intervene and provide them with the resources or guidance they need to be able to

succeed. Educational institutes can use data gathered through the years to predict how students will perform, if they are likely to not graduate, or even if they are likely to change majors. This information can be used to mentor students, for example, by helping students pick courses that may be most useful to achieve their goals, or by giving them advice in terms of the correct major they should be in. To the best of our knowledge, extensive research has not been done in identifying these students, especially when it comes to predicting changes of majors.

We propose modeling students' institutional data in order to predict: i) whether a student is likely to not graduate, and ii) whether a student is likely to change majors. Both of these have a direct impact in the academic careers of students as students who are struggling one way or another could be provided with better feedback, mentoring, or other resources. On the other hand, they also directly benefit the teaching institutions as these predictions could be used to increase retention.

We use institutional data such as the term they entered the university, how many and which classes they have taken, what grades were obtained, whether they have been on probation, whether they have changed major, and their demographics. We then build several models to predict whether they would go on to withdraw (whether voluntarily or by being expelled) or change majors. We also explore how soon into their academic programs we can make these predictions with a high precision. Empirical evidence shows we can be successful at identifying both types of struggles. We present this work in Chapter 4.

2 MODELING ONLINE COURSE DISCUSSIONS

The use of microblogging in classrooms has proven to be a useful and effective way of communication in large classes. In participating in course discussions during class, students are able to ask questions about the material without interrupting the class. This is particularly important if these questions going unanswered could hinder their understanding of the rest of the lecture. Additionally, in very large classrooms, many instructors would prefer for students to not ask their questions whenever they arise and rather have an allocated question and answer time during the lecture. These are some reasons why having a back-channel discussion technology is appropriate. Moreover, some of these technologies allow students to ask questions anonymously, which encourages more questions from the students. In some cases, the students are also able to address other students' questions, thus deepening their understanding and also clarifying doubts other students might have. Additionally, instructors are able to see what questions students have during the class without necessarily disrupting the pace or flow of the lecture.

Given these benefits, the use of a technology that allows a backchannel discussion during class seems ideal. However, deciding to adopt such technology is not an easy decision to make. Many instructors might fear that by adopting such technology the instructors will have an increased amount of workload they might not be able to keep up with, given that most of the times the instructor (or teaching assistant) is expected to answer these questions. This could be a cumbersome activity as the posts pile up and there is only limited time during the lecture to review and address these questions. Another reason to be reluctant in this adoption of technology is that the students might get distracted by noisy or irrelevant posts and not pay much attention to the lecture.

In this chapter we address two problems that are detrimental to the adoption of these technologies. These are: i) the extra time it requires to manage the technology, and ii) the distractions caused by students posting irrelevant content while using this technology. We propose and discusses various approaches to address these challenges and make the use of educational technology more effective for carrying out course discussions. The rest of this chapter is organized as follows:

2.1 Predicting Relevancy of Course Discussions

We develop a novel solution for predicting the relevance of a question asked in a class of a current semester by looking at the questions asked in previous semesters and how they relate to the course lecture material. We also use a set of features from the questions such as the number of students' votes the question received, the number of replies it received, and the length of the question itself. To identify similar questions asked previously, topic modeling and feature selection are used. Topic modeling finds latent, or unobserved topics and terms of interest for the context of the course. Empirical results show that topic modeling leads to better prediction accuracy as compared to feature selection. The similarity of the question and its corresponding lecture material further improves the relevancy prediction of the questions.

The relevancy of a post could be defined in two ways: it could be relevant to the overall course, or it could be relevant to a specific lecture. For example: if the lecturer is discussing credit cards and a student asks a question about insurance, should this be considered relevant or irrelevant? In order to address both definitions, we include features from a course centered relevancy (using topic modeling of data from previous semester) as well as a lecture centered relevancy (using the similarity between the post and the current lecture).

2.1.1 Related Work

Microblogging has been increasingly used as a tool for communication between students and instructors in classrooms over the recent years [8–11]. Some of these works also analyze microblogging in the context of language learning [8]. The use of Twitter, being one of the most popular microblogging service prevalent these days, has also been extensively studied [9,10]. In addition, assessing the credibility of tweets has been investigated in [12].

However, a key problem when microblogging is used in a classroom setting is that of dealing with the overwhelmingly large number responses from the students for a single question posted by the instructor. This clearly grows with time and there is a desperate need for methods to select the best questions to respond. Cetintas et al. [13] provides some approaches using the correlation between questions to identify the most relevant and irrelevant questions. In [14], Cetintas et al. proposes a text categorization approach which can automatically identify relevant and irrelevant questions asked in a lecture by utilizing multiple types of evidence including question text, personalization, correlation between questions themselves and students' votes on questions. In addition, the effect of eliminating stopwords on categorization accuracy is also investigated.

Other related work focuses on finding related questions given a particular question and analyze questions and answers once the user provides a new question [15–18]. [19] proposes a syntactic tree matching approach to identifying similar questions in community-based question-answering services such as Yahoo! Answers. There are methods like [20] that also consider answer quality derived from the expertise of answerers in selecting answers. [21] evaluates the various scoring measures used in finding similarity of tweets and proposes a composite technique that combines the various approaches by scoring tweets using a dynamic query-specific linear combination of separate techniques. However, these approaches are not directly applicable related to answering questions in a classroom setting, particularly because showing

Table 2.1.: Examples of relevant and irrelevant posts

Relevant Posts	**Irrelevant Posts**
So basically options are like insurance policies for shares?	is there an alternate date for the exam 2
what is considered unsecured debt	where is the lecture for today 3/31
How much money should you invest in a bond?	Invest in bacon
Is having no credit worse then having bad credit?	Is insurance more expensive for zombies?

different content to different students based on their interests is not acceptable. Song et al. make use of the users' interest in the questions to identify the most relevant questions [22] and in addition also recommend the most useful questions to users by considering criteria like users' votes. But, unlike Cetintas et al. these methods do not make use of any personalization or correlation among questions at all. In addition, the use of lecture materials, which could be a greatly valuable source in determining relevance, is not explored.

In [14], Cetintas et al. consider a personalization feature, which means that they calculate a student relevancy score based on the relevancy of the questions those students tend to ask. This proved to be effective for their work. However, since we are using previous semesters of data to predict the relevancy of future posts, it is not likely that a student would repeat the course, and if she or he does repeat it, this would likely affect the outcome.

2.1.2 Data

The data used for this study are part of a Personal Finance course from Purdue University. The data used consisted of eight semesters of a course which uses a classroom response system. The data collected consists of questions asked by students during the lectures, as well as outside the lecture time.

Some examples of the types of questions asked are shown in Table 2.1. In this table we show questions labeled as relevant and irrelevant. These questions were manually labeled. If we look closely at the irrelevant posts in the table, we can see that some are clearly irrelevant – *i.e.*, the first two posts; those which contain irrelevant terms

like “exam” and “lecture”. However, the last two irrelevant posts in the table contain terms that could be considered relevant (namely, the terms “invest” and “insurance”) and at first glance might seem relevant. These posts were probably intended as jokes, and should not be considered relevant. This gives us an intuition for why we should not just look at posts that contain relevant terms.

The data collected also includes attributes to the questions, such as the number of votes and replies a particular question received. Even though we could have used the text from the replies, we only considered the number of replies a question received. We also consider whether o not a question was posted anonymously. This could be an important feature especially when the course deals with controversial or sensitive topics.

Apart from the students’ posts, we also used the set of lecture materials provided by the instructor. This set of documents provide the correct context for the course. The lecture materials consist of 21 lectures, out of which the first one contains a mix of logistics and course content, and the rest talk solely about the course content. During the first lecture the instructor discusses the structure of the course, the evaluation criteria, and gives an overall introduction to the course.

Using the findings of [14], all the text was stemmed and stopwords were removed. Stemming is a technique commonly used in information retrieval which reduces terms to their stems. For example, terms such as “walk”, “walks”, and “walking” would all be reduced to their stem term “walk”. Stopwords are terms which are frequently used and do not provide context. Examples of these terms are: “the”, “of”, “for”.

2.1.3 Models

This section we describe the models used to identify the most relevant questions asked in a classroom. We use several logistic regression classification models which were trained using various sets of features. For purposes of training and testing the models, the data were divided into two parts in time, which means that the train data

corresponds to previous semesters, while the test data belongs to future semesters. Within both of these sets, 5-fold cross validation was used by randomly selecting a subset of posts.

Post Features

The first model built uses features from the posts. These features are: the length of the post, the number of votes the post received, the number of replies the post received, and whether or not the post was posted anonymously. These features are referred to as *Post Features* and the model that only uses these features is called *LR_Post*.

It is worth mentioning that this dataset also contained an attribute called "Featured". This attribute allows the instructor of the course to label a post as "featured". This attribute was not considered a feature of the models since only about 1% of the posts were considered "Featured" and so it was heavily skewed. For this reason we do not use this attribute in our models.

Topic Modeling

In order to find which posts are relevant and which ones are not, we first find what topics the students are talking about. The intuition behind this is that we might find that some posts are about topics directly related to the course, while other topics are regarding projects, assignments, exams, etc. We used Latent Dirichlet Allocation, or LDA, to find a set of latent topics given the text of the posts. LDA is a generative model that finds groups or clusters based on the co-occurrence of terms. In LDA, each document is viewed as a finite mixture of an underlying set of topics [23]. We use two approaches to topic modeling, described as follows.

Table 2.2.: Example of topic clusters generated using LDA

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
stock	insurance	class	fund	quote	money	think	company	credit	more
buy	health	exam	mutual	go	account	know	trading	card	need
price	life	question	security	interest	retire	people	nyse	score	one
sell	car	due	social	bond	out	really	information	debt	make
options	company	project	invest	rate	ira	good	difference	debit	such

- Model using LDA:

  The output of the Latent Dirichlet Allocation algorithm is a set of topics with the probability distribution of each post belonging to said topics. For example, let us imagine we have three latent topics, one related to credit, another related to investing, and the third related to insurance, and a post "Is having no credit worse than having bad credit?". The probability distribution for this post might look something like <1, 0, 0>. On the other hand, if we have the post "Do credit cards provide insurance?", the probability distribution might look something like <0.5,0,0.5>. These probability distributions are used as features for this model, along with the *Post Features* discussed in Section 2.1.3. We call this model *LR_LDA_Post*. Depending on how many topics we choose as a parameter for this modeling, we will have that many additional features for this model. The choice of number of topics was selected empirically and is discussed in Section 2.1.4.

- Model using feature selection:

  Another approach to topic modeling we explore is to take the most popular terms of each topic and only consider those terms, disregarding all other less common terms belonging to the topics. For the top terms of each topic, we can find the term frequency of the term in the post. We then have a set of features which are the term frequencies of these terms in the posts. We call this method Feature Selection because we are reducing our feature space by selecting the features that would contribute the most to the model. The idea behind this approach is to keep only the terms that best describe the topics. For example,

let us consider the use of 10 topics with 5 terms in each in the example of Table 2.2. This table shows an example run of the LDA algorithm with the 5 most common terms in each of the 10 latent topics generated from our data. We would have features for the term frequencies of the terms *stock*, *buy*, *price*, etc. We call this model *LR_FeatSel_Post*. In this case we would have 50 additional features.

Lecture Material Features

An important factor when considering the relevancy of a post is what was actually being discussed during that particular lecture. It could happen that the post is relevant to the overall course, but not relevant to the current lecture. In order to take this into consideration, the similarity of the post to the lecture was calculated. The Kullback-Leibler divergence, or KL divergence, was used.

The KL divergence - also known as the relative entropy, is a measure of how different two probability distributions (over the same event space) are. The KL divergence of probability distributions P, Q on a finite set S is defined as:

$$D(P||Q) = \sum_{x \epsilon S} P(x) log \frac{P(x)}{Q(x)} \tag{2.1}$$

It should be noted that this measure is asymmetrical and hence is not strictly a distance metric. Over the past years, various measures have been introduced in the literature generalizing this measure and to make it symmetric. We use one of these symmetric Kullback-Leibler divergence *i.e.*, the Kullback-Leibler Distance (KLD) measure as:

$$D(P||Q) = \sum_{x \epsilon S} \left((P(x) - Q(x)) log \frac{P(x)}{Q(x)} \right) \tag{2.2}$$

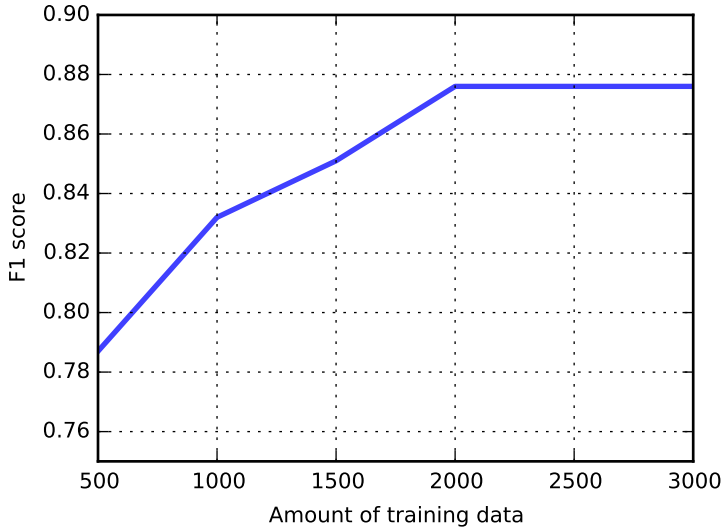

Figure 2.1.: Evaluation of models with various amounts of training data

A problem here is that the lecture text is much larger than the posts, and additionally, each lecture might discuss several topics, some of which are related to the post and some of which are not. In order to solve this problem, we find the KL divergence of the post to the lecture by splitting the lecture text into smaller chunks, or windows. The KL divergence was calculated for each of these windows. The smallest of these divergences (*i.e.*, the most similar), was taken as the divergence of the post to that lecture as a whole. The intuition behind this is that a post might be relevant to only a small portion of the lecture, and this way we can find that portion.

2.1.4 Experimental Results

To understand the efficacy of our proposed models, we ran them on the data described in Section 2.1.2. This section presents the findings of our experiments and addresses some other questions, such as how much training data is representative enough to obtain accurate results and how to obtain the best classification threshold, as well as our choice for window size for the topic modeling.

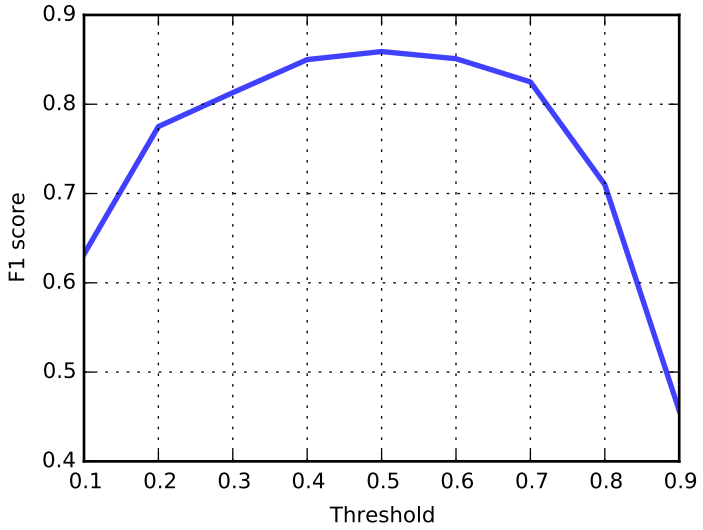

Figure 2.2.: Evaluation of different probability thresholds for predicting the relevancy of posts

How Much Training Data Is Required?

In order to evaluate how much training data is required and enough to accurately be able to model the data, we built the model using various sizes of training data. The sizes were varied from 500 to 3000 data points, increasing every 500 data points. In all cases, the test size was the same: the last portion in time of the data.

As shown in Figure 2.1, the performance of the models increases with the amount of training data, as expected. The accuracy score stabilizes at around a data size of 2000 data points. Considering these results, we use 2500 data points for our experiments.

Threshold Selection

After the models were built and predictions were made for the test data, different cut-offs or thresholds were used to classify a post as relevant or irrelevant. For example, if the threshold were selected to be 0.5, then a post which is predicted to have a

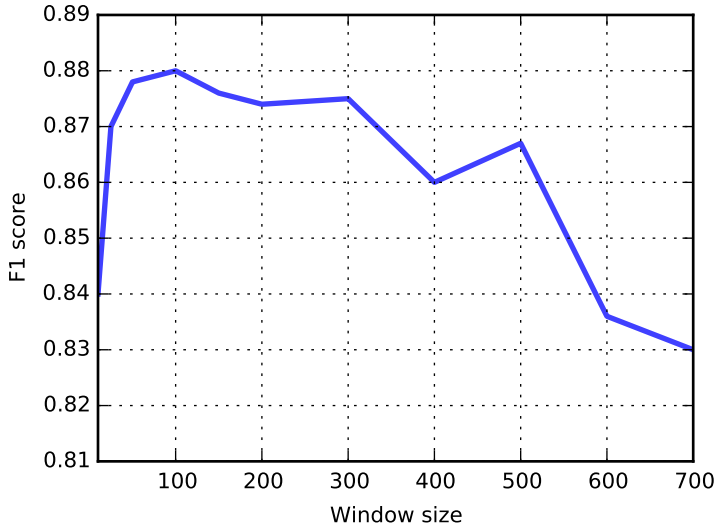

Figure 2.3.: Evaluation of the KL Divergence model using various window sizes

probability of relevancy of 0.6 would be considered to be relevant. Similarly, a post with a probability of relevancy of 0.4, it would be considered to be irrelevant.

The thresholds were varied from 0.1 to 0.9, with 0.1 increases. The accuracy was calculated for each threshold. The threshold that produced the best accuracy was considered the best threshold. This experiment is shown in Figure 2.2. The threshold that yields the best accuracy is 0.5, therefore, this is the threshold used for the rest of the experiments.

Window Size Selection

In order to find an appropriate window size for calculating the KL divergence, we used several window sizes. The windows were made to overlap. This means that if we take the window size to be 100 characters, then the first window would be from 0 to 99, the second one from 50 to 149, the third from 100 to 199, and so on. This experiment is shown in Figure 2.3, which shows that the window size that yields the best accuracy is 100 characters. This might be due to the fact that the maximum allowed size for the posts was set to 140 characters for this course. In this experiment,

100 characters is the closest window size to the length of the posts. The KL divergence of the post to the lecture using a window of size 100 characters was used as a feature, and is referred to as *KLD100*. *KLD100*, together with *Post Features*, form another model called *LR_KLD_Post*.

Evaluation

The evaluation metric used is the F_1 score. The F_1 score is the harmonic mean of precision and recall.

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall}, \tag{2.3}$$

where precision is the fraction of retrieved instances which are relevant, and recall is the fraction of relevant instances which were retrieved. In our experiments we consider all instances which were classified as relevant to be "retrieved".

Precision and recall are evaluation measures that are typically used in information retrieval applications. However, it should be noted that in this case, a combination of both measures is important. It would be trivial to achieve a recall of 100% by simply classifying all posts as relevant, or achieving a high precision by being very conservative and only classifying those instances we are "sure" to be relevant. This approach would not be satisfactory for our application.

Varying the Number of Topics and Terms

Since we do not know how many latent topics there are in this corpus, we experimented by varying the number of topics used for the LDA model. When building the feature selection model it is important to also know how many terms should be used for each latent or underlying topic.

In Table 2.2 we show an example of the top terms of the LDA algorithm when using 10 topics with 5 terms in each. As we can see in this example, some topics are

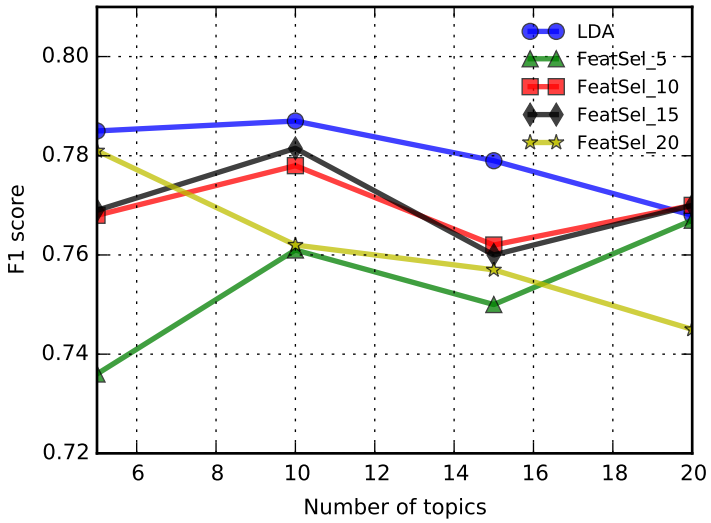

Figure 2.4.: Performance for topic models when varying the number of topics and terms

clearly relevant. For example, topics 1, 2, 4, 5, 6, 8, and 9 seem to be relevant. Topic 3 is irrelevant as it is about course logistics and not directly related to the course content. We can also observe some other topics which are neither obviously relevant nor obviously irrelevant, which are the case of topics 7 and 10.

Figure 2.4 shows several versions of the feature selection model, or *LR_FeatSel_Post*, when varying the number of terms per topic from 5 to 20, and the effect on accuracy on these curves when we also very the number of topics. All the curves include the *Post Features*. The feature selection curves represent the *Feature Selection Features* for the specified number of terms per topic. For example, *FeatSel_05* represents the feature selection features calculated using 5 terms per topic. The curves for feature selection are compared in this figure to the LDA model, or *LR_LDA_Post*, for which the topics were also varied using the same range. At first impression, it might seem there is something inconsistent with this graph. The curves for *LR_FeatSel_Post_5_terms* and *LR_FeatSel_Post_20_terms* seem to have an almost opposite behavior. Now, let us see what is happening here. When the number of topics is 5, *LR_FeatSel_Post_5_terms*, which has 5 terms, will have at most 25 terms as features. This is a low number of

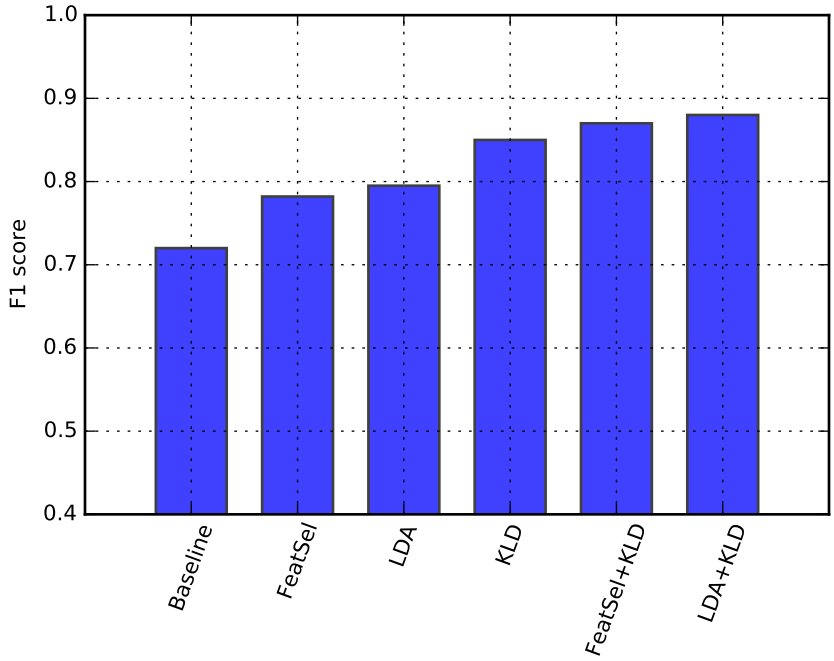

Figure 2.5.: Model comparison for predicting the relevancy of posts

features to capture the whole context of the course, and therefore would explain the low accuracy. This is not the case of *LR_FeatSel_Post_20_terms*, as it starts off with at most 100 terms when the number of topics is 5. However, when the number of topics is 20, it has at most (because duplicates are removed) 400 terms. This large number of features could explain the lower accuracy for this model, since it is most likely adding terms that are not representative of the context.

Based on the results of Figure 2.4, we chose 10 topics with 15 terms for the *LR_FeatSel_Post* model and 10 topics for the *LR_LDA_Post* model.

Discussion of Model Performance

The different models explored and described in section 4.3.2 were compared and the results are shown in Figure 2.5. All the different models shown in this figure in-

clude the *Post Features* described in section 2.1.3. The model shown as *Basic* contains only the *Post features*, *i.e.*, model *LR_Post*. Following this model, we show the models *LR_FeatSel_Post*, *LR_LDA_Post*, and *LR_KLD_Post*. Then we show the models which include the post features together with topic modeling features and KLD features, *i.e.*, *LR_FeatSel_KLD_Post* and *LR_LDA_KLD_Post*. From this figure we can see that having the *Post Features* alone (*i.e.*, model *LR_Post*) yields the lowest F_1 score of 0.72. Adding the feature selection features to the former (*i.e.*, *LR_FeatSel_Post*) gives us an F_1 score of 0.782 (an improvement of 8%), while adding the LDA features to it (*i.e.*, *LR_LDA_Post*) achieves an F_1 score of 0.795 (an improvement of 9.7%).

Comparing the two topic modeling approaches to the KL divergence approach, we can see that the KL divergence performs better. With the *LR_KLD_Post* model we obtain a F_1 score of 0.850, which is an improvement of 17.4% over the model with only *Post Features*. It should also be noted that including the *KLD100 Feature* to both topic models achieves a better performance. When the *KLD100 Feature* is added to the Feature Selection model (*i.e.*, *LR_KLD_FeatSel_Post*), the F_1 score goes up to 0.870, which is a 11.3% increase over the *LR_FeatSel_Post* model. Similarly, when the *KLD100 Feature* is added to the LDA model (*i.e.*, *LR_KLD_LDA_Post*), the F_1 score of the model goes up to 0.880, which is a 10.7% increase over the *LR_LDA_Post* model.

Some of these findings were published in [24].

2.1.5 Conclusions

In this study we explored several techniques to classify educational microblog questions as relevant or irrelevant. These techniques use features from the nature of the questions, such as their length, the number of replies, number of votes, and their anonymity; features extracted from topic modeling on questions from previous semesters; and features extracted from the similarity of the questions to the lecture materials. All these features are shown to be helpful in predicting the relevancy of

questions. LDA performs slightly better than feature selection for our application. We also show that adding the similarity (or divergence) of the posts to the lecture material further improves the performance of the techniques.

By predicting the relevancy of questions asked in a classroom, we are able to promote the most relevant quesitons so that the instructor can quicky identify and address the most important questions that need answering. Furthermore, by promoting the most relevant questions, we help student not get distracted by irrelevant content. Although, not studied in this dissertation, it is also our hope that by students seeing more relevant content, they would be more motivated to ask relevant questions.

2.2 Ranking Posts by Relevancy and Diversity of Topics

In Section 2.1 we discussed some solutions to minimize the amount of irrelevant content in online course discussions by identifying the most relevant posts to the lecture content, thus saving the instructor valuable time in reviewing the questions, and minimizing the distractions caused by irrelevant content. However, when identifying the most relevant content and promoting it there is the problem of promoting repeated questions. Many students have the same or very similar questions, and while many systems (including Hotseat) have a voting mechanism for this purpose, in an in-class setting we cannot expect the students to go over all the questions to see if their question was already asked.

To minimize the amount of repeated questions, while still considering which questions are the most relevant to the lecture content, we propose a different take on this problem. We view it as a ranking problem in which we consider: i) the similarity of the question to the lecture material, and ii) the diversity of topics in the questions. For the latter, we use a submodular algorithm which calculates the gain or diminishing returns in terms of diversity of topics, that the question would add by being included. For example, let us imagine we have two very similar questions about

credit cards which score a very high relevance score and one question about interest rate which does not score as high as the other two questions. In this case, we should first promote one of the questions about credit cards and then, because including the question of interest rates gives us a higher diversity of topics, we should include the question regarding interest rates even though it has a lower relevance as calculated by our algorithms. Using our framework, we can sort the posts by their relevance and diversity and thus show the most pertinent content to the students.

The following are two key aspects of this problem:

- Relevance: We want to recommend relevant questions that will match the course material.
- Coverage: The recommended relevant questions should cover all aspects of the lecture by keeping the recommended set of posts as diverse as possible.

Our proposed algorithms address both these aspects.

2.2.1 Related Work

[14] proposes a text categorization approach which can identify relevant and irrelevant questions asked in a lecture by utilizing multiple types of evidence including question text, personalization, correlation between questions themselves and students' votes on questions. However, it does not explore using topic modeling, nor does explore using submodularity to mitigate the problem of repeated questions. The use of topic modeling for microblog content has been explored in [25].

[22] makes use of the users' interest in the questions to identify the most relevant questions [22] and in addition also recommend the most useful questions to users by considering criteria like users' votes. However, including the lecture material, which could be a greatly valuable source in determining relevance, is not explored.

Submodularity appears in a wide range of application areas including social networks and viral marketing [26] and document summarization [27]. The most relevant

(a) Word cloud of text from lecture material

(b) Word cloud of text from posts

Figure 2.6.: Word distributions of lectures and posts

work to our submodular framework are [28–30]. Our work differs from [28, 30] in which it is assumed that there is a click model available. However, in our application we do not have access to click data. Our work is similar to [29], however instead of using blog posts, we focus on short questions or comments.

2.2.2 Data

As the previous section, this study uses data collected using Hotseat, a microblogging classroom response tool developed at Purdue University. The same eight

semesters of data of a course on Personal Finance were used. The data consists of questions or comments posted by students during the lectures.

Figures 2.6a and 2.6b show word clouds for a particular lecture material in our dataset and its corresponding posts from students. Some examples of questions asked are shown in Table 2.1. In this table, we show some questions manually labeled as relevant and irrelevant.

Apart from the students' posts, we also used the set of lecture material provided by the instructor. This set of documents provides the correct context for the course.

2.2.3 Framework

In this section we discuss the models that comprise our framework as well as the different feature representations used to model the data.

Feature representation

The data described in Section 2.2.2 can be represented in various ways and each of these ways would impact the modeling significantly. For example, we might use a representation that only takes into account the terms used; or the terms used and the frequencies of these terms (*i.e.*, Term Frequency, or TF), or the terms used, their frequencies in the question and their frequencies in the whole collection (*i.e.*, Term Frequency – Inverse Document Frequency, or TF-IDF); or the terms and the order in which they occur; or a representation based on language modeling. We select several representations to determine the best representation for this application and discuss them in this section.

- Unigrams: The text of the document (in this case, question) is represented as a set of words and their frequencies. The order of the words is disregarded. This is one of the simplest representations, yet commonly used for text categorization. The terms are represented in a Bag of Words (BOW) representation.

- Bigrams: There are many terms that might be relevant or not depending on the context. The use of bigrams is an effort to solve this problem. Bigrams are two terms that are used together. For example, the word "credit" might be used in a relevant context if we are talking about "credit card", but it could also be used in an irrelevant context if we are talking about "extra credit". For this study, we first find all bigrams for each post and represent them in a BOW representation.

- Topic Modeling: Topic Modeling is used in our study to find the distribution of topics in the posts as well as the distribution of topics in the lecture material. The intuition behind this is that we might find that some posts are about topics directly related to the course, while others are about topics regarding projects, assignments, exams, *etc.* We use Latent Dirichlet Allocation, or LDA [23], to calculate these distributions. LDA is a generative model that finds groups or clusters based on the co-occurrence of terms by assuming that each document $d \in D$ is associated with a K-dimensional topic distribution. In other words, each document d covers K latent topics, where each topic is defined as a distribution over words drawn from a Dirichlet distribution $\phi_k \sim Dirichlet(\beta)$. Elements of ϕ_k denotes the probability that a particular word is used for that topic. We use Online LDA [31], an implementation of LDA which uses variational inference instead of a Collapsed Gibbs Sampler[1] for practical purposes. After calculating this, we have a distribution of probabilities that indicate the probability of a post, or lecture, belonging to each topic. The dot product of the two distributions is calculated and used as a similarity metric for this model.

  Applying LDA on short documents such as tweets, text messages, or microblog questions, is challenging. Previous efforts mainly focused on tweets where researchers applied methods including aggregating all the tweets of a user into a single document [33] which follows an author-topic model. However, this model

[1]We used Gensim Python Library [32] for the model estimation process, also available at `https://pypi.python.org/pypi/gensim`

fails to capture the fact that each tweet has its own topic assignment. Latest approaches such as Twitter-LDA [34] tried to overcome this issue; however, it assumes that a single tweet is usually about a single topic which conflicts with our assumptions that a post is about multiple topics with different layers. Labeled LDA [35] is another LDA-based approach, however this model relies on labeled data such as hashtags, thus it might not cover the topics the user mentions. In our application we do not have any explicit signal such as hashtags on the posts, which makes the model inherently inapplicable to our case.

Relevancy

In order to make sure we are presenting content that is relevant to the course material, we compare each post to the lecture material provided for the course. We use cosine similarity as the similarity measure. Since the lengths of the posts and the lecture material are very different, we split the lecture material into smaller chunks and compute the similarity between the post and each chunk, as we did in Section 2.1. The intuition behind this is that the post will likely be relevant to a portion of the lecture and not to the lecture in its entirety. A sliding window is used to divide the material in smaller chunks. For example, for a window size of 50, if the first chunk goes from characters [0, 49], the second chunk goes from characters [25, 74], and the third, from characters [50, 99], and so on. As mentioned before, the lecture material and the posts are stemmed and stopwords are removed.

Submodularity

Submodularity is a discrete optimization method that shares similar characteristics with concavity while resembling convexity. Submodular functions exhibit a natural diminishing returns property, *i.e.*, given two sets S and T, where $S \subseteq T \subseteq V \setminus v$, the incremental *value* of an item v decreases as the context that v is considered grows from S to T.

More formally, submodularity is a property of set functions, *i.e.*, the class of functions $f : 2^V \rightarrow R$ that maps subsets $S \subseteq V$ to a value $f(S)$. V is a finite ground set, often referred as the *ground set*. In our application, V refers to the posts that are being asked in a given lecture. The function f maps any given subset to a real number. The function f is called normalized if f($\emptyset$) = 0, and it is monotone if $f(S) \leq f(T)$, whenever $S \subseteq T$. The function f is called submodular if the following equation holds for any $S, T \subseteq V$:

$$f(S \cup T) + f(S \cap T) \leq f(S) + f(T) \tag{2.4}$$

An equivalent definition of the submodularity property is as follows:

$$f(S \cup \{v\}) - f(S) \leq f(R \cup \{v\}) - f(R) \tag{2.5}$$

where f is submodular if for any

$R \subseteq S \subseteq V$ and $v \in V \setminus S$. This form of submodularity directly satisfies the property of diminishing returns; the *value* of v never increases when the context gets larger.

It has been shown that submodular function minimization can be solved in polynomial time [36], however, submodular function maximization is an NP-complete optimization problem and intractable. However, [37] shows that the maximization of a monotone submodular function under a cardinality constraint can be solved near-optimally using a greedy algorithm.

In submodular function maximization, we are interested in solving the following optimization problem:

$$A^* = \underset{A \subseteq V : |A| \leq k}{\operatorname{argmax}} f(A) \tag{2.6}$$

subject to a cardinality constraint k. If a function f is submodular, takes only non-negative values, and is monotone, then even though the maximization is still

Algorithm 1 Greedy submodular function maximization with budget constraint

Require: V,k
Ensure: Selected set of posts S
1: Initialize $S \leftarrow \varnothing$
2: **while** $|S| \leq k$ **do**
3: $\quad v \leftarrow \text{argmax}_{z \in V \setminus S}(f(S \cup \{z\}) - f(S))$
4: $\quad S \leftarrow S \cup \{v\}$
5: **end while**
6: **return** S

NP complete, we can use a greedy algorithm (see Algorithm 1) to approximate the optimum solution within a factor of $(1 - 1/e) \approx 0.63$ [37].

We integrate submodularity in our study by using the BOW representation for each post as well as for the whole lecture material. We follow a greedy approach of the post that maximizes the gain by having its terms included in the set of "posted" terms. For example, let us assume the BOW of the lecture material L and posts P_1, P_2, and P_3 are:

$L : \{stock : 2, market : 1, company : 1, insurance : 3, life : 2\}$

$P_1 : \{stock : 1, market : 1\}$

$P_2 : \{life : 1, insurance : 1, company : 1\}$

$P_3 : \{stock : 1, company : 1\}$

For this example (and following step 3 of Algorithm 1), adding P_2 first yields the highest gain, as it covers three terms from the lecture material. Step 4 of the algorithm is to include that post to the set of selected posts S. Then, by removing the already included terms, we can update the BOW for L as:

$L : \{stock : 2, market : 1, insurance : 2, life : 1\}$

We then cover the largest amount of remaining terms by including P_1 next since it adds two terms (*i.e.*, the highest gain of all posts), and then P_3 which adds just one term.

Next, we introduce our submodular framework using the notion of *generating functions* introduced by [28], which can be embedded into a large family of submodular functions.

Definition 2.2.1 *A monotonic, concave, and non-negative function* $\sigma : [0, \infty) \rightarrow [0, 1]$ *is a cover generator.*

Theorem 2.2.1 *Given a domain* X *and coefficients* $c_x \geq 0 \; \forall \; x \in X$*, and* $A \subseteq X$*, the function* $\rho(A) \doteq \sigma(\sum_{x \in A} c_x)$ *is submodular whenever* σ *is a cover generator.*

Proof ρ is a monotone function with respect to addition to A. Because of concavity in σ and due to $c_x \geq 0$ for all $x \in X$, one can see that ρ is submodular. [28] ■

We use a *probabilistic cover* as σ function as proposed in [28], defined as:

$$\sigma(z) = 1 - e^{\theta z} \text{ for } \theta > 0 \tag{2.7}$$

Next, we define our submodular framework as follows. L is the "source" content which represents the lecture to be covered, P is the list of posts chosen to represent the content in lecture L, w_j is a real value describing the extent to which feature j is present in the post p. Then, our submodular framework is defined as follows.

$$f(L, P) = \sum_{l \in L} \sum_{j} w_j \rho_p(P) \tag{2.8}$$

Since our algorithm is submodular, it provides a $1 - 1/e$ approximation which can be maximized with a greedy algorithm. The algorithm starts with an empty summary, which is what we call the subset of "selected" posts to represent the lecture. In each step, a post is added to the selected post list that results in the maximum relative increase of the objective. The algorithm terminates when a predefined budget k is reached.

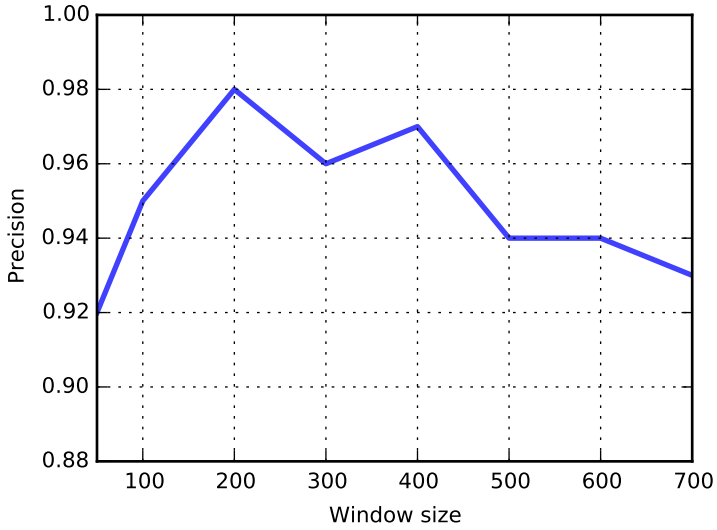

Figure 2.7.: Relevancy precision for different lecture window sizes

2.2.4 Experimental Results

To demonstrate the effectiveness of the proposed solutions, we ran experiments on the data presented in Section 2.2.2. In this section we present our findings. To provide a base case for comparison, we compare the proposed solutions with time based sorted posts, which we will refer to as the baseline.

As was done in the Section 2.1, in order to select a window size for dividing the large lecture materials into smaller chunks, we split the lecture in chunks that varied in size from 50 characters to 700 characters and measure the effect of chunk sizes on precision. Figure 2.7 shows the results. A window size of 200 is used for the rest of the experiments as it yields the best results. Next, we conduct an experiment to understand the best feature representation for the posts.

Using the relevancy model, we compare the effect of unigrams, LDA, and bigrams on the precision of the top-k ranked posts. The results are shown in Figure 2.8. The figure shows that the unigram representation outperforms bigrams and LDA representations. Interestingly, LDA performs the lowest of the three representations. A possible reason for this is that the text is short, which might make it harder to tell

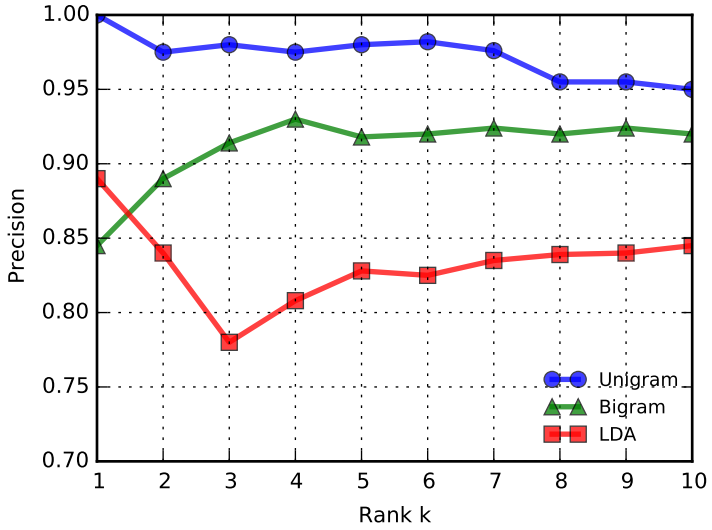

Figure 2.8.: Precision vs rank using relevancy model for different feature representations

the topic(s) of the posts. The reason for unigrams to perform better than bigrams might be that for bigrams the stopwords were not removed, in order to account for expressions that might be lost if we remove the stopwords. Since unigrams performs the best, we use this feature representation for the rest of the experiments.

In the next experiment, we compare the different models (*i.e.*, relevancy alone, submodularity alone, relevancy with submodularity, and the baseline which is a time based sorting) on their performance for measuring relevancy. The results are shown in Figure 2.9. The two models that use submodularity outperform the model that uses only relevancy, which in turn, significantly outperforms the baseline. Both models that use submodularity perform similarly well, with the submodularity alone model performing better in the first 6 ranks, and the model that combines submodularity and relevancy performing better in the lower ranks.

Figure 2.10 shows that the recall was very similar for all our experimental models, with their recall values all being slightly better than the baseline model.

Next, we analyze the diversity of the posts using the models. Diversity is computed as the unique number of terms in the top-k posts. Figure 2.11 shows the results. The

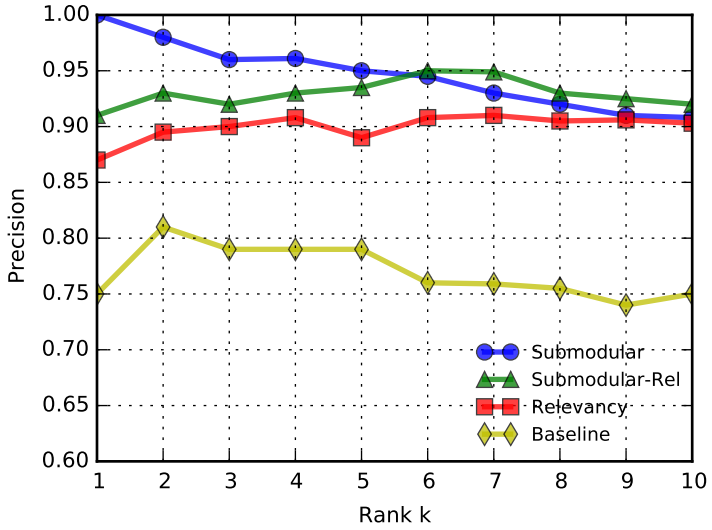

Figure 2.9.: Precision vs rank for different models using unigrams

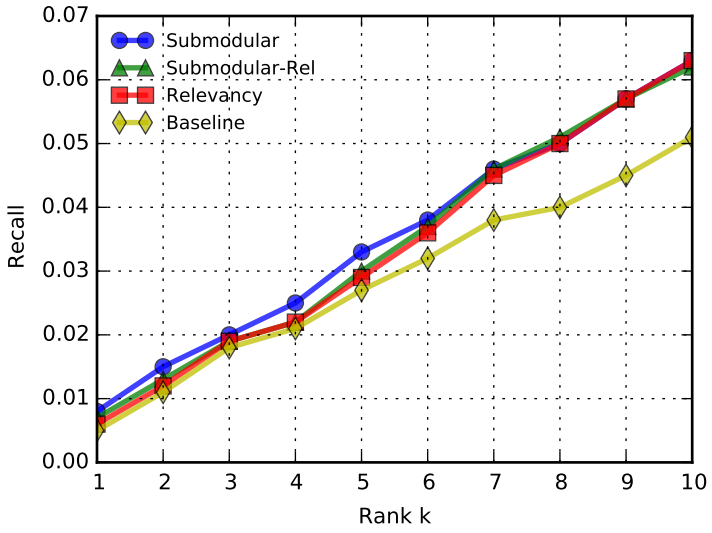

Figure 2.10.: Recall values at different ranks

submodularity models perform better than the relevancy and baseline models. This shows that the submodular function does in fact diversify the top retrieved posts. Interestingly, the diversity in the relevancy model is lower than that of the baseline. This shows the inability of the relevancy model to provide enough coverage of topics.

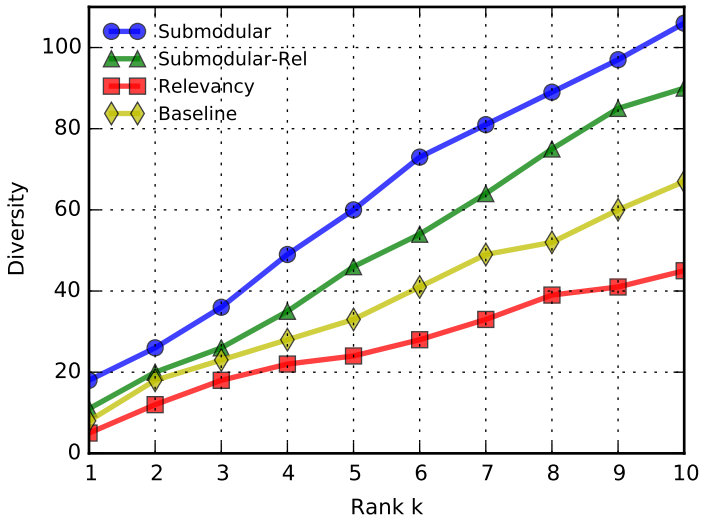

Figure 2.11.: Diversity values at different ranks

Overall, we see that the proposed models are able to identify diverse and relevant content with high precision. Some of the findings in this section were published in [38].

2.2.5 Conclusions

In this chapter we proposed a novel framework to identify the most relevant and diverse content in educational online discussions. The proposed framework addressed three main problems that occur in these scenarios: (i) high amount of irrelevant content, (ii) repeated or similar content, and (iii) overwhelming amount of posts, thus the need for finding the important questions in a very limited time. We proposed solutions that rank the posts based on their relevance to the lecture material. We also proposed a model that would diversify the topics mentioned in the top ranking posts so as to cover different aspects or topics of the lecture, thus reducing the amount of repeated questions.

Our empirical analysis shows the effectiveness of our proposed framework. In particular, we show the increase in diversity, while not compromising on the effectiveness of predicting relevancy, when adding submodularity. By ranking the questions by

their relevance to the lecture and by the gain of diversity coverage the questions offer we can effectively promote the most important questions, thus reducing the number of irrelevant and repeated questions at top ranks. This not only reduces distractions, but also saves the instructor valuable time.

3 MODELING INTELLIGENT TUTORING SYSTEMS' CONTENT

Intelligent tutoring systems, or ITS, are systems aimed at providing students with instructions that adapt to their knowledge acquisition, as a human tutor would do. ITS are particularly beneficial at being able to deliver one-on-one personalized teaching without the need of a teacher for each student, and have been shown to be as effective as their human counterparts [5]. These ITS, however, rely on having each problem and subproblem labeled with the skills or knowledge that would be required in order for students to be able to solve them.

When designing exercises within an adaptive learning software, appropriate knowledge components should be assigned to them. As defined by [6] and [7], a knowledge component, or KC, is a description of the simplest mental structure, process, or step that a learner uses, alone or in combination with other knowledge components, to solve a task or problem. KCs assigned to exercises are a prerequisite for performing many types of analyses on educational data. Some of the data mining approaches that are often applied are: prediction, clustering, relationship mining and distillation of data for human judgment [39]. The process of assigning KCs to the exercises can be a time consuming job, since the number of possible KCs can be very large. In order to help the tutor or course designer in writing exercises, this chapter discusses two approaches for the suggestion of KCs for untagged content in ITS. The first approach is based on text mining [40] and an SVM classification algorithm and the second is based on a search engine with language modeling [41] and a kNN classification algorithm. These two approaches can be used for a system that could encourage the course designers to assign knowledge components to new exercises they design, as well as to existing exercises that do not have knowledge components assigned to them.

For this study, we use data gathered from the ASSISTments platform [42]. The ASSISTments platform is a web based tutoring system, developed at Worcester Poly-

technic Institute with the collaboration of Carnegie Mellon University, that assists students in learning by adapting the questions according to the pace of the students. It also provides teachers with an assessment of their students' progress. The system started in 2004 with a focus on 8^{th} grade mathematics, in particular helping students pass the Massachusetts state test. It has since expanded to include 6^{th} through 12^{th} grade math and scientific inquiry content. A feature that sets ASSISTments apart from other systems is its robust web based content building interface [43] that is designed for rapid content development by system experts and teachers alike. Teachers are responsible for a growing majority of the content in ASSISTments. While the content has been vetted and verified as being of educational value by ASSISTments system maintainers, the content often lacks meta information such as KC tagging as this is an optional step in content creation. An ASSISTments administrator must add this tagging or leave it blank, which can cause a lack of accuracy in student model analyses of the data and also inhibits the system from reporting KC information to teachers. The tagging has to be performed by selecting from the large list of KCs, which are organized into 5 categories and sorted alphabetically within the categories. Untagged content in ASSISTments is a growing phenomenon with only 29% of the content possessing KC tags as of this writing. Accurate KC suggestion would expedite the processes of content tagging and encourage external content builders to tag their content.

The necessity of associating KCs with problem solving items is shared by a number of tutoring systems including The Andes physics tutor [44], The Cognitive Tutors [45] and the ASSISTments Platform [43]. The Andes and Cognitive tutors use student modeling to determine the amount of practice each individual student needs for each KC. The student model that these tutors use is called Knowledge Tracing [46], which infers student knowledge over time from the history of student performance on items of a particular KC. This model depends on the quality of the KC model to make accurate predictions of knowledge. Poorly tagged or untagged content can deprive

the model of crucial information needed to make inferences about students knowledge state.

3.1 Related Work

This work continues the line of research proposed in Rose *et al.* [47] and expands on the prior art by applying a variety of optimizations as well as evaluating the algorithms on numerous KC models of varying granularity. [47] presented KC prediction results on a model of 39 KCs but skill models have since increased in complexity. We investigate how KC prediction accuracy scales with larger KC models and which algorithms adequately meet this challenge.

The KC association with items in a tutor is typically represented in an $Item \times KC$ lookup table called a Q-matrix [48]. Methods such as Learning Factors Analysis [49] have been proposed to automate the improvement of this Q-matrix in order to improve the performance of the student model. Recently, non-negative matrix factorization methods have been applied in order to induce this Q-matrix from data [50]. While the results of this work are promising, its applications thus far are limited to test data where there is no learning occurring and only to datasets with around five KCs, where these KCs represent entirely different high level topic areas such as Math and English which rarely intersect. To the best of our knowledge, all the student modeling and Q-matrix manipulation methods have thus far not used any information of the text of the items they are evaluating. Our work makes the contribution of looking at this source of information for making accurate KC predictions. While our work focuses on text mined KC suggestion to aid content developers, this technique is relevant to those interested in Q-matrix improvement as well.

3.2 Data

The dataset used for testing the performance of the proposed approaches was taken from tagged content on the ASSISTments platform during the 2005-2006 school year.

Table 3.1.: Intelligent tutoring system sample questions with corresponding knowledge component labels for various granularity models

Question text	5 KC label	39 KC label	106 KC label
Look at the figure. What is the measure of angle x?	Geometry	Line-intersection-angle-formation	Transversals
Lee correctly answered 5 out of 20 questions on the test. What percent of the questions did Lee get correct?	Patterns-Relations-Algebra	Number-representations	Finding-Percents
What is the area of the shaded part of this figure?	Measurement	Measurement-formulas-and-techniques	Area-of-Circle
When 7 is subtracted from a number four times, the result is 3. What is the number?	Number-Sense-Operations	Setting-up-and-solving-equations	Equation-Solving

The ASSISTments platform has three KC models consisting of varying degrees of granularity. The first two models, containing 5 and 39 KCs, use KC names corresponding to the Massachusetts state math standards. The systems finest-grained KC model contains 106 KCs which were created in-house [51]. The KCs from the 106 model have a hierarchical relationship to the 39 KC and 5 KC models. This allows content to be tagged only with the 106 KCs and then inherit the KCs from the other models in the hierarchy. Table 3.1 shows some examples of questions from our dataset and the corresponding KCs associated with them for each of the granularity models. While tagging with the 106 model is preferred, content builders can choose from KCs from any model to tag their content.

The dataset is composed of 2,568 exercises which were tagged with one or more KCs. 80% of the exercises are tagged with a single KC, 15% are tagged with two, and 5% are tagged with between 3 and 5 KCs. For exercises with more than one KC, only the first KC was kept. The exercises in the dataset can be very diverse, containing different number of sentences, just words, or only mathematical symbols. Some exercises contain images, which are represented with <img> tags. These images, however, were not used as part of this study.

The distribution of knowledge components is not uniform, which means that some KCs occur more frequently than others. The distribution of the KCs in the 106 model is shown in Figure 3.1. The x-axis shows the top 30 occurring KCs, while on the y-

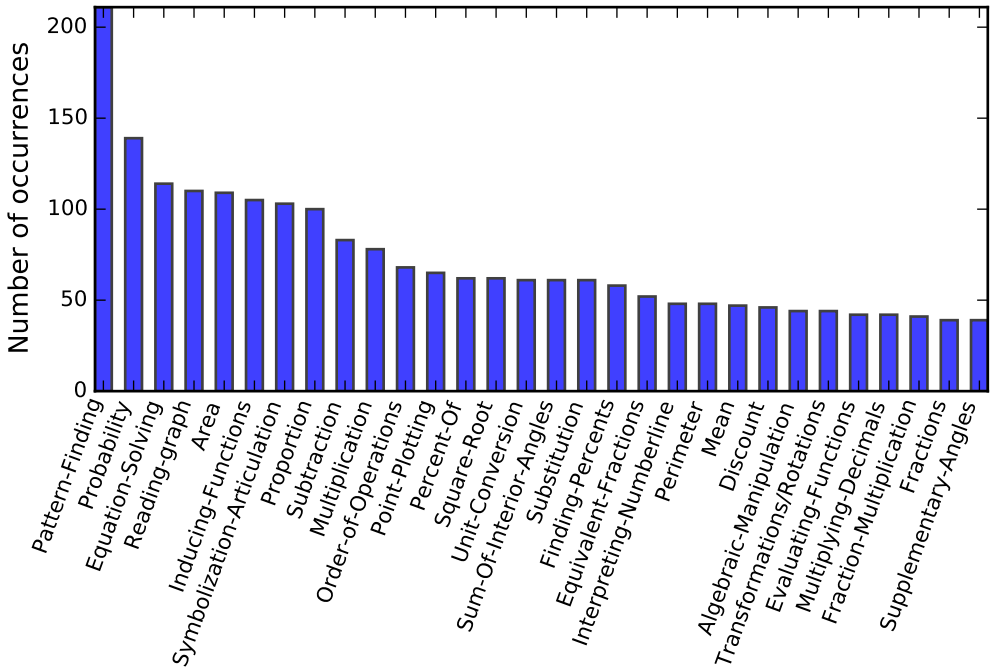

Figure 3.1.: Distribution of knowledge components in the dataset

axis shows their frequencies in the data samples. The figue shows how the KCs are not equally distributed. If we were to show all the KCs (106 in total) in the figure, it would follow a similar distribution with a long tail. This distribution of KCs is such that 50% of the content is tagged with 20% of the KCs.

3.3 Framework

In this section we describe our proposed solutions for assigning appropriate KCs by providing automatic suggestion system in the process of exercise design. Our first approach consists of text mining and an SVM classifier, and our second approach consists of language modeling using a search engine based approach with a k Nearest Neighbors (kNN) classifier.

3.3.1 Text Mining with SVM Classifier Approach

One approach was based on text mining and building SVM classification model using the Text Garden [52] utility. It has been shown [40] that the SVM is an appropriate method for text classification. The main reasons include the ability to handle high dimensional input space and suitability for problems with dense concepts and sparse instances. For our experiments, we use a One-Against-All multi-class SVM model. The classification model was built based on the set of labeled exercises.

In adition, we want to test the influence of stop words removal and stemming on the classification problem, therefore four different classification models were built, covering all combinations of applying these standard text processing techniques.

3.3.2 Language Modeling and kNN Approach

The second approach was to use the Lemur Toolkit [41], which uses the Indri search engine, an open source search engine based on language modeling. The Indri search engine uses a query language that allows documents to be indexed and queried. This approach was used together with a k Nearest Neighbors (kNN) classification algorithm. kNN is a commonly used algorithm that finds the k documents closest (*i.e.*, most similar) to the document being tested. We use a set of train questions with contain labels (*i.e.*, the KC associated to them), and test questions which we want to predict the KC for. We use the Indri search engine to find the most similar questions to our test set questions and assign the most likely KC by association.

Within Indri, we use a query likelihood model with a multinomial unigram language model. This model, which derives from Bayes' Rule, predicts the likelihood that a query is observed as a random sample from the document language model. For our application, we use the questions from our train set as our document collection. We construct a language model for each document in the collection using multinomial unigram language models and then use the text from the questions in our test set as queries. Our model then finds the likelihood, for each document, that the query

Table 3.2.: Example retrieval of similar questions for suggesting knowledge components for untagged math problems

	Text	KC
Query	When 7 is subtracted from a number four times, the result is 3. What is the number?	?
Similar retrieved problems	1. If you subtract 6 four (4) times, how much, in total, are you subtracting?	Equation-Solving
	2. You are subtracting 28 from a number and you are left with 3. What is the initial number?	Equation-Solving
	3. We can find the shaded area by finding the area of the square and then subtracting the area of the circle. Lets start by finding the area of the circle.	Area

was generated by the language model of that document, following Equation 3.1, and ranks all documents by this likelihood.

$$P(Q|M_d) = \prod_q P(q_i|M_d)^{\frac{1}{|Q|}} \tag{3.1}$$

Using this search engine we are able to retrieve the top k, where k is chosen to be 200, most relevant search results (*i.e.*, the top k of the ranked likelihoods, which are most relevant to the query), along with their KC tag. Each retrieved document is then assigned a score based on its rank, where the top retrieved document (*i.e.*, the most similar one), would have the highest score. For instance, if $k = 200$, the score of the top retrieved document is 200; the score of the second retrieved document is 199, and so on.

$$tag_{score}(t) = \frac{a}{\sum score(t) + b} \tag{3.2}$$

We then calculate a score for each possible tag as defined in Equation 3.2, where $\sum score(t)$ is the summation of all document scores with tag t, and a and b were both chosen to be two times the KC model size. This is done to predict KCs using a weighted measure of the frequencies of tags (*i.e.*, KCs) and their retrieval ranks.

Table 3.3.: Experimental results for suggesting knowledge components for untagged math problems

	Number of suggestions									
	SVM					kNN				
Dataset	1	2	3	4	5	1	2	3	4	5
106 KC	0.607	0.739	0.784	0.809	0.823	0.574	0.736	0.796	0.835	0.865
106 KC ST	0.621	0.749	0.798	0.824	0.842	0.567	0.728	0.795	0.834	0.866
39 KC	0.683	0.815	0.863	0.895	0.914	0.666	0.815	0.854	0.898	0.914
39 KC ST	0.689	0.818	0.870	0.901	0.916	0.653	0.829	0.865	0.907	0.914
5 KC	0.814	0.943	0.969	0.981	1.000	0.762	0.919	0.976	0.993	1.000
5 KC ST	0.815	0.938	0.969	0.983	1.000	0.784	0.923	0.976	0.996	1.000

Lastly, for each test question (query), the tag with the highest tag_{score} is assigned to it.

Table 3.2 shows an example set of questions that could be retrieved for a given query (*i.e.*, a quesiton for which we want to predict its KC label). In this example, the KC with the highest score is "Equation-Solving".

3.4 Experimental Results

Testing was performed using the 5 folds cross validation method. All experiments used sensitivity, or recall, as the goodness metric, since we are interested in evaluating the likelihood of getting the correct labels in the top k results retrieved. For both approaches, testing was performed on the three different knowledge component models: the largest model with 106 KCs, the 39 KC model, and the 5 KC model. In [53], the automatic text generation of mathematical word problems is performed. The paper shows that leaving out the common text processing techniques, namely stop word and stemming, can increase the performance of text categorization. To take into account the findings of that paper, we tested each dataset with four different text processing setting: (1) without applying stopwords removal and stemming; (2) applying only stop-words removal; (3) applying only stemming; and (4) applying both stop-words removal and stemming.

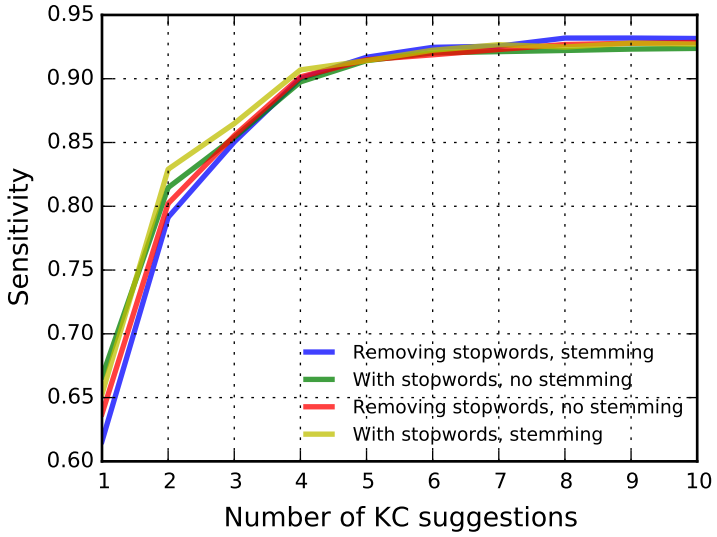

Figure 3.2.: Sensitivity of the kNN classifier with different preprocessing methods for the 39 KC model

The experimental results of both approaches (*i.e.*, the SVM method and the query retrieval method with kNN) are shown in Table 3.3. The table shows sensitivity results given KC suggestions ranging from 1 to 5. Sensitivity, in this case, represents the likelihood that the correct KC is suggested among the top k suggestions for the exercises. The sensitivity when suggesting 5 KCs, *e.g.*, is the percentage of exercises where the correct KC was among the top 5 suggested KCs. For the 5 KC model, 5 suggestions always results in 100% sensitivity. Each row represents different datasets (with various KC granularities) and different text preprocessing settings used in the experiment. 106KC, 39KC and 5KC are labels for the different knowledge components models. SVM and kNN are labels for the two different classification algorithms. ST indicates applying stemming. Each column of the table represents different number of suggestions. Table 3.3 shows that the SVM classifier with stemming outperforms the kNN model when the number of suggestions is between 1 and 3. If we show more suggestions, then the kNN model tends to perform better. The performance of stopwords removal did slightly worse than stemming. The performance of stopwords removal in addition to stemming also performed slightly worse than just stemming.

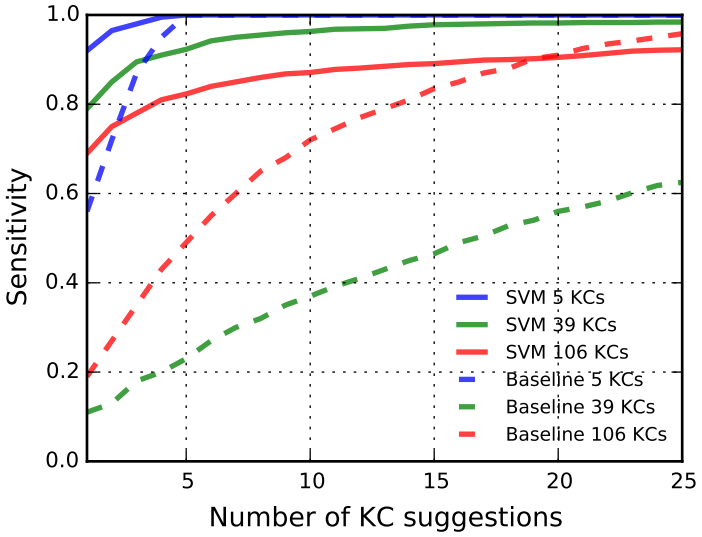

Figure 3.3.: Sensitivity of the SVM classifier on different KC granularity models

Figure 3.4 shows the results for the kNN model on the 39 KC model using different preprocessing methods (*i.e.* the four stemming and stopword removal combinations). This figure shows that for the first 5 suggestions, leaving the stopwords and stemming is slightly better than the other methods. For the subsequent 5 suggestions (*i.e.* from suggestions 5 to 10), removing stopwords and stemming seem to outperform the rest of the combinations. However, these improvements (in both cases) are much less significant than in the referenced paper. The improvement for the best options in comparison with the worst option, *i.e.*, removing stop-words and not stemming, was around 2%.

Figure 3.3 shows the SVM model performance for all models of KC granularity. These curves are shown in comparison to our baselines, which are the suggestion of the most common KCs. This makes sense as a baseline since the KCs are not evenly distributed. This figure shows that with a small number of suggestions we can achieve a high sensitivity score. It also shows the drastic improvement of each KC granularity model compared to its baseline. Each one of the curves for our solutions significantly outperforms its baseline curve, which proves the effectiveness of our SVM models.

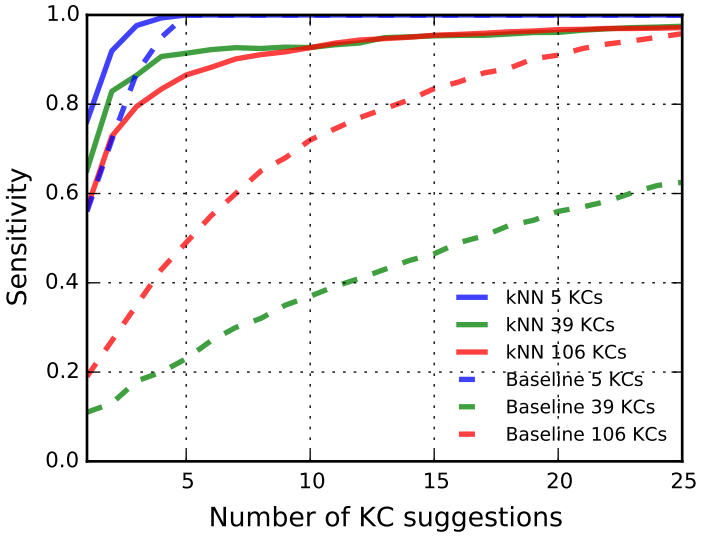

Figure 3.4.: Sensitivity of the kNN classifier on different KC granularity models

Figure 3.4 shows the kNN model performance for all models of KC granularity in comparison to our baselines. This figure also shows the significant improvement of using our models over the baselines and how we are able to reduce significantly the number of labels the content creators would need to navigate through to make their selections.

Figure 3.5 compares the sensitivity results for both of our approaches. This figure shows an interesting result. When working with the two more coarse datasets (*i.e.*, 5 KC and 39 KC models), the SVM model outperforms the language model with kNN. However, for the most fine-grained dataset, which is also our most important dataset, the kNN model outperforms the SVM model. This suggests that the language model with kNN is able to differenciate between more specific topics. It is also a possibility that the SVM model is suffering from class confusion.

Some of the findings in this section were published in [54].

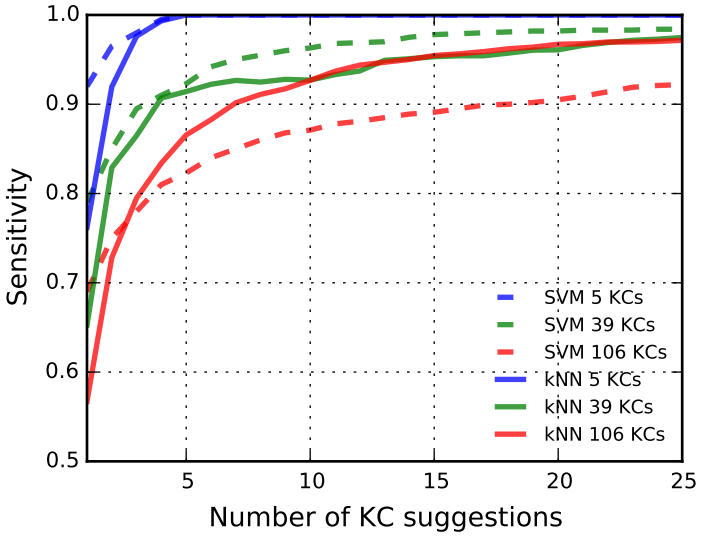

Figure 3.5.: Sensitivity comparison of our classifiers on different KC granularity models

3.5 Conclusions

In this chapter we proposed a framework for suggesting knowledge components for untagged content in intelligent tutoring systems. Tagging all problems with their corresponding knowledge components is a tedious task that requires time and energy to complete. Many times it is not completed properly, thus negatively affecting the functioning of the tutor algorithms. Our results indicate that both suggested approaches are suitable for practical usage, since they would decrease significantly the number of KCs to be used for labeling, without compromising much on performance (*i.e.*, failing to show the correct labels). This would not only save the content creators a great amount of time and effort, but would likely make the ITS algorithms function more efficiently as less content would be left unlabeled.

The dataset with 106 KC model is the hardest challenge for the proposed approaches, yet this is the KC model for which the system can be mostly useful in practical applications. If only one KC is suggested for the 106 KC model, it would be the correct one in 62.1% of the cases. Suggestion systems usually suggest more than

one option so as to increase the likelihood of the correct option being amongst the suggestions. If the number of suggestions is 5, the correct KC would be among these 5 in 86.6% of the cases, or in 92.7% of the cases if there are 10 suggestions provided. If the number of suggestions increases, the probability that the correct KC is among them naturally grows, however, the effort required from the user to choose the correct KC among the suggested also increases. Yet, our results show that we can greatly minimize the number of KCs the user would need to select from by showing them 5 to 10 suggestions, and in the great majority of the cases this would satisfy their needs.

4 MODELING STUDENT SUCCESS

When it comes to higher education, many students struggle trying to graduate. There is a need to identify these students both for the students' benefit as well as for the benefit of the academic institution. Students invest much of their time, energy, and money pursuing their degrees. There is an abundance of reasons why students struggle, and finding some of these could be the difference between them leaving the university empty handed or leaving with a degree. Moreover, if universities are able to identify the factors that contribute the most to students not graduating, they could come a step forward in improving retention, which would be greatly beneficial for they internal planning. With institutional data from students such as their cohort, their academic program, demographic data, how many classes the students are taking, how well they did, and other information such as their academic standing, we have a great opportunity of predicting students' performance. This information could be used by mentors to better understand students' challenges, or to give these students early warnings of how they seem to be performing. For example, departments could reach out to students who seem to be at-risk of not graduating and offer direct guidance, or they could be matched with students who are performing well in a mentoring program.

In addition to identifying students who would go on to drop out or be expelled, there are students struggling because they are in the wrong academic program. Rather than struggling for a few more terms, these students probably need better guidance. It is for this reason that we also study whether we can identify students who would go on to change major.

In this chapter we propose various models to assist the university in retaining students. In particular, we propose models to predict i) if a student is at risk of not graduating, presented in Section 4.3, ii) how soon we can we identify they are at risk,

presented in Section 4.4, and iii) if a student is likely to change majors, presented in Section 4.5.

4.1 Related Work

In the past few decades, the use of academic analytics and course management systems has increased significantly [55]. According to [56], the use of academic analytics for higher education can be summarized in four categories: finance, grant management, advancement, and student services, the latter being the most frequent use. Within the student services area, the most common uses are for student recruitment or enrollment management and for retention. The work reported the most common uses of enrollment management being (in order of frequency): identifying potential students who are the strongest prospects for admission, alerting officials when enrollment metric falls outside a desired range, forecasting future demands for courses, and tailoring recruitment strategies for individual prospective students. The most common uses for retention purposes were (ordered by frequency): identifying students who might be at risk academically, and alerting officials when an academic intervention with a student is warranted. This work also presents benefits of academic analytics for institutional outcomes, the most popular being: improved fund-raising results, improved admission/enrollment results, improved the institution's financial results, and improved student retention results.

Some influential studies of student retention include the work in [57], which studies student persistence and retention and argues that students remain at an institute depending on how they perceive themselves to be doing. Another influential study in [58] argues that much of student retention has to do with the involvement of students with faculty and peers.

[59] studied integrations of students in a course management system to predict academic success. Some CMS integrations used in the study include: viewing content files, reading messages, reading discussion postings, posting discussion items, viewing

calendar entries, posting calendar entries, among others. The study found 10 CMS features which are significantly correlated to student success and was able to correctly identify nearly 60 percent of students needing help. The study, which used undergraduate students' data, found that if they focus on freshmen' data the results are more significant. In this variant, it was able to correctly identify nearly 80 percent of students needing help.

Purdue University developed Signals [60], an application which detects early warning signs of students at risk of not graduating. Signals predicts student performance based on grades, demographics, past academic history, as well as students effort as measured by interaction with Blackboard Vista, Purdues learning management system. Signals allows instructors and advisors to send emails to the students at risk and provides pre-written messages for a more effective communication. In [61] it is reported that there is a significant increase in retention of students using Signals for one of their courses when compared to students not using it in any of their courses. Similarly, there is a significant increase in retention of students using Signals in two or more courses compared to those using it in only one.

In a qualitative analysis carried out at Purdue University in Arnold *et al.* [62], it is reported that faculty and administration believe that with Signals's early intervention and their feedback to students, Signals's provides an important support for students. They also report an increase in student success and retention.

There has been some work done regarding changes of majors and their relationships with graduation, GPAs obtained, as well as how it relates or impacts the student's motivation. [63] presents a study carried out from data from multiple universities in which the impact changing majors has on graduation is analyzed, with an emphasis on STEM majors. It concludes that the association between changing majors and withdrawing depends to an extent on the student's gender, entering declared major status, the timing of the change, and various other student-level and institutional-level characteristics. In [64], the relationship between declaring a major

and motivation was studied. It found that the longer the student takes to declare a major, the lower the motivational indicators of the student.

To the best of our knowledge, the problem of predicting whether a student will change majors has not been explored extensively in the past. [65] studies the prediction of change of majors with data from students from Carnegie Mellon University. Their study emphasizes on the timing of the declaration of majors. This study was also conducted in a much smaller university than the one we use, with only 6 majors to choose from.

4.2 Data

We use registration data from Purdue University. This dataset consists of student demographics (self-identified ethnicity, age, gender, belonging to a minority group), which semester they joined the university, registration information per term (academic period, student level, current college, current major, classes registered in, number of credits enrolled in, current academic status, full or part-time status),grades obtained per term (courses they completed, grades obtained, number of credits earned, GPA), and graduation information (whether they graduated, voluntarily dropped, or were expelled).

From this data, we extracted a set of 19 features to use in our modeling:

`gender` - *Binary value* indicating the student's gender

`minority` - *Binary value* indicating whether the student identifies as a member of an underrepresented group

`full-time` - *Binary value* indicating whether the student was studying full-time at some point

`probation` - *Binary value* indicating whether the student was in probation at some point

`probation-twice` - *Binary value* indicating whether the student was in probation for at least two terms

`avg-credits-attempted` - *Numeric value* indicating the average number of credits attempted per term

`avg-credits-earned` - *Numeric value* indicating the average number of credits earned per term

`avg-credits-failed` - *Numeric value* indicating the average number of credits failed per term

`avg-credits-gpa` - *Numeric value* indicating the average number of gpa credits earned per term

`gpa-credits` - *Numeric value* indicating the number of credits being used for GPA calculation

`quality-points` - *Numeric value* indicating the total number of quality points the student has earned[1]

`gpa` - *Numeric value* indicating the latest GPA of the student

`num-terms` - *Numeric value* indicating the number of terms the student has enrolled in

`change-college` - *Binary value* indicating whether the student has changed college at some point

`change-college-higher-grad-rate` - *Binary value* indicating whether the student has changed to a college with a higher graduation rate than the previous college the student was in

`change-major` - *Binary value* indicating whether the student has changed major at some point

`last-college-grad-rate` - *Numeric value* indicating the graduation rate of the last college the student was in

`last-major-grad-rate` - *Numeric value* indicating the graduation rate of the last major the student was in

`last-major-avg-gpa` - *Numeric value* indicating the average GPA of all the students in the same major

[1]This is the number of credit hours multiplied by the grade index for each class

4.3 Predicting Student Success: Identifying At-Risk Students

Early warning systems aim at identifying students who are at risk of not graduating. With the large amount of institutional and data on students' academic progress, higher ed institutions have the capability of identifying these students and improve retention. As pointed out by Campbell in [59], improving retention is of importance to the students, the institution, and society. From a student's perspective, completing a higher ed degree will provide them with more job opportunities and an increase in salary. Additionally, with the rise in degree expectations from employers the importance of completing a higher ed education can be easily understood. From the institution's perspective, this work notes that, "retention rates are often regarded as important indicators of institutional quality and commitment to undergraduate education. Institutional retention rates impact public perception, institutional reputation, future enrollments, and faculty morale." From the society's perspective, the work points out that with a more educated society, we will have increased tax revenues, decreased reliance on public assistance programs, lower unemployment rates, among other possible benefits.

We define the problem of predicting whether students will withdraw as a supervised binary classification problem. We have two classes: graduation and withdrawal (whether voluntarily or not). The labels we use for our classification problem are whether or not the students withdrew from their current academic program (whether voluntarily or not). That is, we use a 1 if they withdrew (as this is what we want to predict) and a 0 it they graduated. For this study, we remove the students who are still studying, as we want the model to learn which traits will result in a higher or lower likelihood of withdrawing and for these cases we do not have labels. Secondly, we remove all data that occurred after the student either graduated or withdrew, as a student could graduate from one program and then join another and withdraw or vice versa, thus producing conflicting labels.

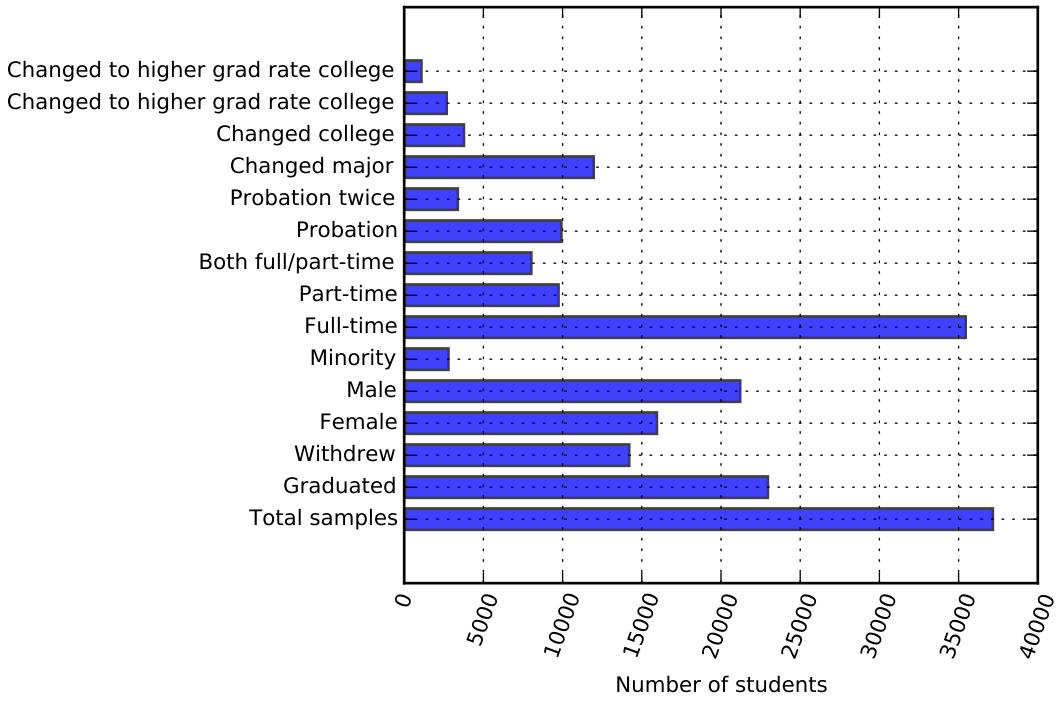

Figure 4.1.: Data exploration for predicting student withdrawal

The rest of this section is organized as follows: 4.3.1 describes the dataset used and the set of features extracted used. Additionally, it provides some data exploration in order to have a better understanding of the data used. Section 4.3.2 presents the models used for the classification problem. Section 4.3.3 presents the experimental results obtained by the models trained, and finally, Section 4.3.5 discusses the findings and concludes Section 4.3.

4.3.1 Features

Figure 4.1 provides us with some context before starting our modeling. It shows we have 37,164 data samples, out of which 14,204 of them withdrew while 22,958 of them graduated. 15,953 students identify as female, while 21,209 identify as male, and 2,792 identify as belonging to an underrepresented group. Furthermore, 3,770 students

Table 4.1.: Correlation between features and student withdrawal and respective p values

Feature	**Correlation to withdrawing**	p **value**
quality points	-0.624234943135	< 0.0001
last major graduation rate	-0.538978426731	< 0.0001
gpa	-0.523590979745	< 0.0001
gpa credits	-0.504342353891	< 0.0001
last college graduation rate	-0.466680645281	< 0.0001
avg credits earned	-0.455278198202	< 0.0001
avg credits failed	0.453064186389	< 0.0001
probation	0.399578299083	< 0.0001
last college graduation rate	-0.388570586012	< 0.0001
number of terms	-0.356210852028	< 0.0001
avg credits gpa	-0.275950599814	< 0.0001
probation twice	0.222828163194	< 0.0001
full time	-0.223098341242	< 0.0001
avg credits attempted	-0.21018610131	< 0.0001
minority	0.0797970949683	< 0.0001
changing major	-0.074641024616	< 0.0001
female	-0.0419017460189	< 0.0001
change college higher grad rate	-0.0347672960694	< 0.0001
change college	-0.0339250512379	< 0.0001

changed college at some point during their academic career and 11,961 students at some point changed major. Out of the students who changed college, 2,683 of them changed to a college with a higher graduation rate than the one they were previously in, and 1,087 of them did the opposite change (*i.e.*, to a college with a lower graduation rate). 35,445 students were full-time students, while 9,741 were part-time students, and 8,024 were both full and part time students at some points. Moreover, 9,921 students were at some point in academic probation, while 3,382 of them were in academic probation for at least two terms.

In order to have a better understanding of the features we should use for the modeling, we first look at correlations between the variables listed in Section 4.2 and whether the students ended up withdrawing from the university, shown in Table A.2. The features are ordered by magnitude and the p value is shown for an $N = 37,162$.

This table shows that the number of quality points a student has earned is the most correlated, in this case inversely, to withdrawing. This feature is followed by the the graduation rate of the last major the student was in, the GPA of the student, the number of GPA credits, the graduation rate of the last college the student was in, and the average number of credits earned, all of these being inversely correlated. The most positively correlated feature is the average number of credits failed per term by the student, followed by whether the student was in probation at some point, and then by whether the student was in probation for two terms.

It should be noted that all the features listed in Table A.2 are statistically significant, with a p value of less than 0.0001, for the sample size. Some other features, which were not as significant, are included in the appendix.

4.3.2 Models

In order to predict whether students are at risk of not graduating (*i.e.*, withdrawing from their program, whether voluntarily or not), we ran several classification models. Since we cannot easily visualize our feature space and we would not know the shape of the decision boundary we need for our classification problem, we build several different machine learning classifiers, namely: Logistic Regression, Support Vector Machines (SVM) with a linear kernel, and Decision Tree. We use these models to classify our data and determine if a particular student is likely to not graduate. All of these models are commonly used in practice and, in general, are known to be highly effective classifiers. All the features described in Section 4.3.1 were used for the three models in order to determine which one is more suitable for this classification problem.

Table 4.2.: Logistic regression confusion matrix for identifying at-risk students

	Predicted graduated	Predicted withdrew
True graduated	4337	254
True withdrew	634	2207

4.3.3 Experimental Results

To evaluate our models, we trained and tested them using the data described in Sections 4.2 and 4.3.1. These models were trained using 80% of the data for training and the remaining 20% for testing, in 5 rotations, *i.e.*, with a 5-fold cross validation.

We evaluate our models based on precision, recall, and F_1 score. For this specific application, we care more about the recall than about precision because the consequences of a type I error (*i.e.*, identifying a student who is not at risk as an at-risk student) are not big. However, not identifying a student who is actually at-risk (type II error), and furthermore, not providing them with the additional support they might need could make the difference between the student graduating or not.

In addition, we present the Receiver Operating Characteristic (ROC) curve for each model and compute the Area Under the Curve (AUC) of our models as another evaluation metric. AUC presents the area under the Receiver Operating Characteristic (ROC) curve, which plots the true positive rate (TPR) against the false positive rate (FPR).

Logistic Regression Modeling

Table 4.2 shows the confusion matrix of the predictions made by the logistic regression model. This table shows that out of the 2,840 (633 + 2,207) students who withdrew in each cross validation fold, we were able to correctly identify 2,207 of them, in average; which gives us a 77.7% recall.

Table 4.3 shows the coefficients of the logistic regression function trained. These coefficients suggest that, for instance, an increase in the graduation rate of the last

Table 4.3.: Logistic regression model coefficients for identifying at-risk students

Feature	**Coefficient**
last major grad rate	-4.41968
changed major	1.01131
last major avg gpa	0.75420
last college grad rate	-0.48504
change to college with higher grad rate	0.48109
probation once	0.37939
minority	0.34175
changed college	-0.29705
num terms	0.29067
probation twice	0.27224
avg credits failed	0.26491
gpa	0.16113
avg credits earned	-0.15512
avg credits gpa	0.14797
avg credits attempted	0.10979
female	-0.04678
full time	0.02187
gpa credits	-0.01801
quality points	-0.01240

major the student was in, would correspond in a decrease in the likelihood of withdrawing. The same can be said about the graduation rate of the last college the student was in, though to a lesser extent. On the other hand, changing major, or changing to a college with a higher graduation rate, as well as being in probation, all correspond to an increase in the likelihood of withdrawing. It also suggests that increases in the average GPA of the last major the student was in, as well as increases in the number of terms the student has been enrolled, being part of a minority group, and increases in the student's GPA, all correspond to increases in the likelihood of withdrawing.

Table 4.4.: SVM confusion matrix for identifying at-risk students

	Predicted graduated	Predicted withdrew
True graduated	4384	207
True withdrew	793	2047

Table 4.5.: Decision tree confusion matrix for identifying at-risk students

	Predicted graduated	Predicted withdrew
True graduated	4125	466
True withdrew	625	2215

Support Vector Machines Modeling

Table 4.4 shows the confusion matrix for the SVM model. Out of the 2,840 (793 + 2,047) student who withdrew in each cross validation fold, the model was able to correctly identify 2,047 of them. This results in a recall of 72.1%.

Decision Tree Modeling

A decision tree model was trained to identify features that might behave differently for some students. From a very simple decision tree trained using our dataset, with depth of three shown in Figure 4.2, we can see that the most influential feature is the number of quality points. For this example, if the number of quality points a student has earned is greater than 138, then the GPA is the next best indicator; while, if the quality points is lower than or equal to 138, then again we look at the quality points as the best indicator. This tree of depth three serves as an intuition behind the model, however, the tree trained and used for this analysis has a depth of 50. While we chose a large depth for our experiments, it should be noted that we used early stopping in order to avoid overfitting.

Tables 4.5 shows the confusion matrix and evaluation of the decision tree model. This table shows that out of the 2,840 (625 + 2,215) student who withdrew in each

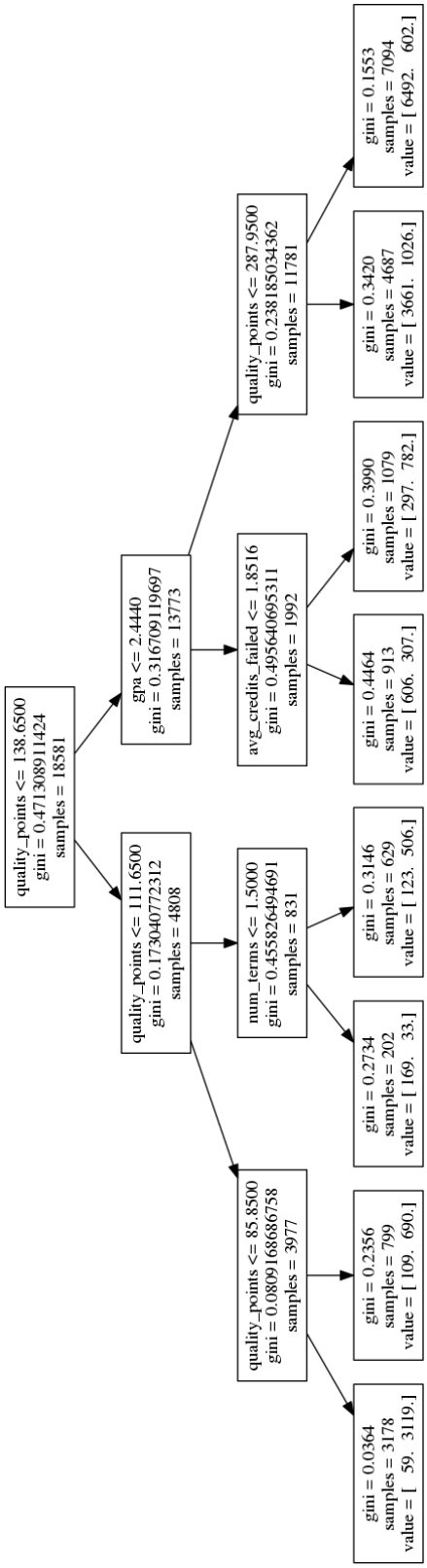

Figure 4.2.: Example decision tree with a depth of three

Table 4.6.: Evaluation of models for identifying at-risk students

	Precision	**Recall**	F_1 **score**	**Support**
Logistic regression	0.896	0.777	0.832	2840
Support vector machines	0.906	0.721	0.802	2840
Decision tree	0.826	0.780	0.802	2840

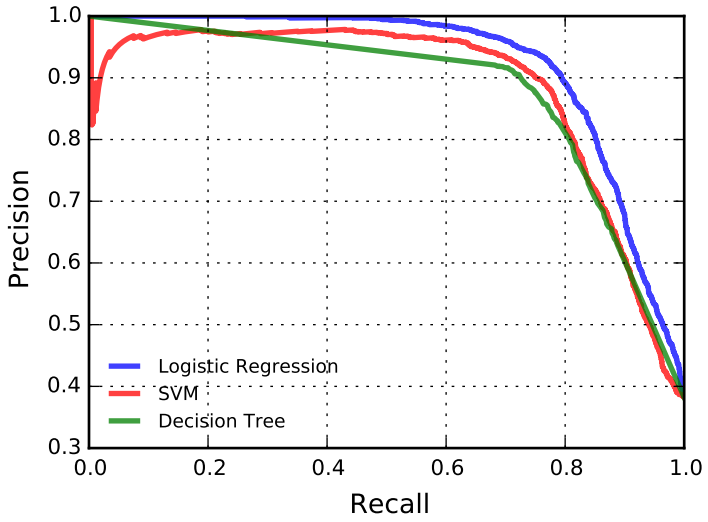

Figure 4.3.: Precision vs recall for each model for predicting withdrawal

cross validation fold, the model is able to correctly identify 2,215 of them; thus we have a 78.0% recall.

4.3.4 Model Comparison

Table 4.6 presents a comparison of the evaluation of our models. Specifically, it shows the precision, recall, and F_1 measure for the different models built. SVM yields the highest precision of the three models, followed closely by logistic regression. On the other hand, when we look at recall, we see the opposite behavior; the decision tree model performs the highest, followed by logistic regression, and then SVM. As far as the F_1 score, the logistic regression model performs the highest of the three models.

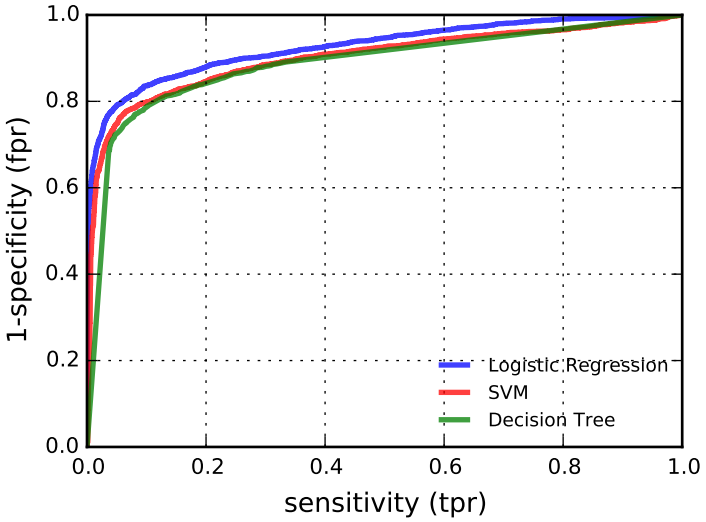

Figure 4.4.: ROC curve for each model for predicting at-risk students

The three models achieved very high results when looking at the precision against recall and the ROC curves, with logistic regression performing the best, followed by SVM, and then decision tree, as we can see from Figures 4.3 and 4.4. By varying the classification thresholds we use in deciding whether a student will withdraw or graduate, we can obtain various sets of precision and recall. More specifically, since at the end of the predictions we have a probability of the sample belonging to each class, we can decide what our threshold for classification is. We can then use different thresholds to plot a recall vs precision curve for each model, shown in Figure 4.3. This curve tells us what precision we would get for a certain recall. For example, using logistic regression, we can identify 80% of the students who are likely to withdraw with a precision of 89%, or identify 70% with a precision of 96%). Using SVM, we would get an accuracy of 83% for identifying 80% of the at risk students and 94% for identifying 70%. With decision tree, we would get accuracies of 81% and 92% respectively, for identifying 80% and 70% of the at-risk students.

Figure 4.4 shows an ROC (*i.e.*, Receiver Operating Characteristic) curve, which is the true positive rate against the false positive rate for different classification thresh-

Table 4.7.: Evaluation of models: AUC values for all models for predicting at-risk students

Model	AUC
Logistic Regression	0.921
Decision Tree	0.882
Support Vector Machines	0.898

olds. This figure shows how the false positive rates and the true positive rates compare for the three models. Again, with this evaluation, we can see that logistic regression model outperforms the other two models, and their specific AUC values are presented in Table 4.7.

There are differences in the models which would make them more or less suitable for this specific application. Since we do not know what the data actually looks like (*i.e.*, what our feature space looks like), or even if it is linearly separable, we need to try different models that take into consideration different data assumptions. In general, the differences in these models lie in the shape of the decision region. For example, if the classes (graduation and withdrawal) are linearly separable with a single decision boundary, logistic regression would be more suitable. If the decision boundaries are parallel to the axis, making a sort of rectangular class region, then decision trees would be suitable. For example, if we have two features (x_1, x_2) then it can only create rules such as $x_1 \leq 2.5$, $x_2 \leq 5$, which we can visualize as lines parallel to the axis. If the data is not linearly separable, SVM works best, as it would map the data to a higher dimension which is separable.

4.3.5 Conclusions

In this section we proposed predicting whether a student was likely to not graduate, by either voluntarily withdrawing or by being expelled. We used various machine learning models to classify students into two classes, namely graduation and withdrawal (whether voluntarily or not). We were able to achieve the best F_1 score, recall,

and AUC with the logistic regression model, which were of 0.91, 0.94, and 0.92, respectively. While these numbers are very high, it is important to note that they are across all terms available for the students. If we are interested in predicting the withdrawal with the intention of early intervention, we must use only partial data as soon as we are able to identify that they are at risk. This brings us to the importance of Section 4.4, where we discuss how early we are able to make these predictions with a decent precision.

It is also important to note that we can aim for an even higher recall (*i.e.*, being able to correctly identify more students who are at risk), at the expense of a lower precision. If we wanted to be able to identify more students who might be at-risk and the academic institution is willing to deal with the consequences of false negatives (*i.e.*, incorrectly identifying students who are not at risk as at-risk students), we can reduce the threshold used in the classification. This means that if we were classifying students as at-risk if the likelihood was greater or equal to 0.5, we could reduce this to 0.4, or 0.3, for example. There is no clear threshold that would be ideal; it depends on how aggressive or conservative the institution wants to be in identifying these students.

Some of the features or variables used in this study are attributes that the students cannot change, such as their demographics. However, of the variables they can change, *i.e.*, the behavioral ones, we identified two that were particularly influent in these predictions. Those are whether or not the student changed major and whether or not the student has been in probation. Both these variables can be indications of other struggles. For example, the student might be changing major for reasons other than academics. Them going on to change major is not going to fix those underlying problems.

4.4 How Early Can We Detect At-Risk Students?

In the previous section we proposed models to predict if a student would withdraw from their academic program or not. However, a crucial part of this question is *how early can we detect a student is at-risk?* As important as identifying at-risk students is, it means very little if we do not provide them with better guidance on time. The sooner we can identify them, the sooner the institution could start to mitigate the risks, which is the whole point of this study. To this end, we want to analyze *when* it is possible to identify these at-risk students with an acceptable precision, in order to provide these students with the resources they need before it is too late. The rest of this section is organized as follows. Section 4.4.1 describes our methodology and the models applied. Section 4.4.2 presents the experimental results obtained and Section 4.4.3 discusses and concludes this section.

In order to determine how early we can start to identify the students who are struggling and are more likely to end up withdrawing from their academic programs, we use the same data and features described in Section 4.3.1 with one modification. We use the data for each student up to a set number of terms. For this purpose, we set the `num-terms` variable and then for each student, we consider the data accumulated up to those terms. For example, for `num-terms`=2, we consider only the first 2 terms worth of data for each student. This way we are modeling with only the information gathered until that point into their academic careers and would know if that is enough time to make accurate judgements.

Similar to how we predicted whether a student was likely to withdrawal from their program done in Section 4.3, we use the following set of 18 features for our modeling. The only feature left out is `num-terms` because it is constant.

Figure 4.5 shows at what point in their academic programs did the students withdraw (whether voluntarily or not) from the university. This figure shows a large number of students withdrawing after their second term. This information is impor-

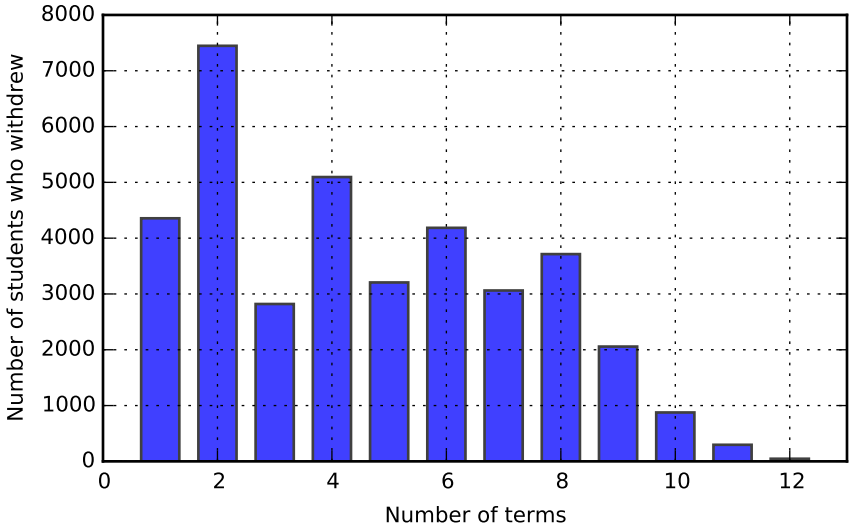

Figure 4.5.: Number of students who withdrew after a given numbers of terms

tant when we model this data as we would like to be able to intervene as soon as possible with these students.

4.4.1 Models

Since our goal is to determine how soon we can accurately predict a student is at risk of not graduating from his or her academic program, we model the data the same way we did in Section 4.3, where we predicted whether they will likely end up not graduating. That is, we run various machine learning classifiers: Logistic Regression, Support Vector Machines, and Decision Tree. This time, however, we run them for several values of the feature named `num-terms`, as described in Section 4.4. The intuition here is that we want to draw cut offs to determine which number of terms is the lowest one that also gives us an acceptable precision.

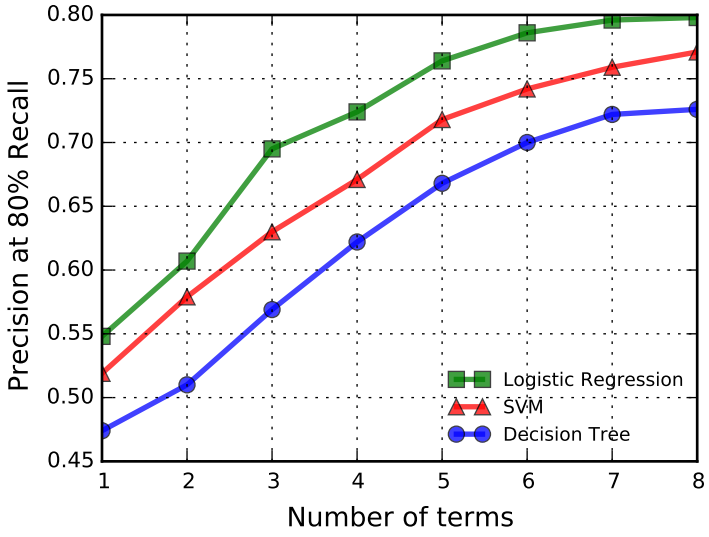

Figure 4.6.: Precision at 80% recall for all models using different numbers of terms

4.4.2 Experimental Results

The curves in Figure 4.6 show the tradeoff between waiting longer to make predictions and an increase in the precision of these predictions. We can see that, consistent with the figures seen before, Logistic Regression continues to outperform SVM and Decision Trees when restricting the data to a set number of terms. Looking at the curve for Logistic Regression with observations of 3 terms, we can get a precision of nearly 70% while identifying 80% (*i.e.*, recall of 80%) of the at-risk students (withdrawing students). However, if we wait two more terms, we can improve the precision to 76%.

Since the logistic regression classifier outperforms the other models studied, we use this model to compare the precision values at different recall values after the students have been studying for a certain number of terms. This comparison is shown in Figure 4.7. this figure shows the tradeoff of recall and precision. If we want to achieve a higher recall (*i.e.*, being able to identify a larger number of at-risk students), we would need to compromise on precision (using one of the lower curves in the figure). This figure shows we can achieve a precision of 95% if our aim is to identify 50% of

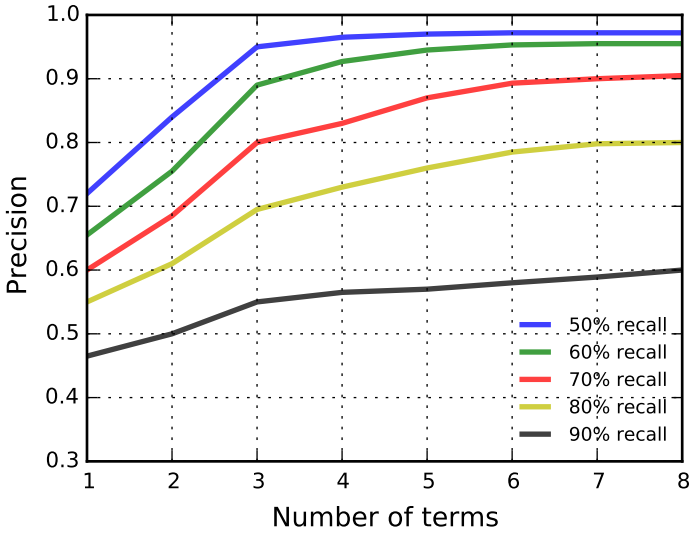

Figure 4.7.: Precision at different recall values after different numbers of terms, using a logistic regression model

at-risk students by the end of their third term studying. On the opposite side of the spectrum, if our aim is to identify 90% of students who are at risk by the end of the third term, we would do so with a precision of 55%. Perhaps an even better balanced tradeoff would be a recall of 70% with a precision of 80%, by the end of their third term.

4.4.3 Conclusions

In this section we aimed at determining how early we can identify at-risk students and at establishing when is the right or appropriate time to intervene and provide the students with the help they likely require. As discussed in Section 4.4.2, we are able to identify a large amount (80%) of at risk students by the end of their third term studying, with a precision of nearly 70%. While the precision for these predictions is not extraordinary, we should keep in mind two things: i) our aim of identifying 80% of the at-risk students is somewhat ambitious; and ii) we care more about recall than we do about precision. In that regard, it is acceptable to reach out to a student who

seems might need help, even if they would end up graduating without reaching out. If we wanted, however, a higher precision, we could for instance, identify 70% of at-risk students with a precision of 80% at the end of their third term. Moreover, we show that if we wait for the students to study a few more terms, our precision significantly improves. However, the longer we wait, the more we risk losing students who needed help and did not receive it on time. Furthermore, because our predictions are in the form of a probability, we could rank them and only reach out to the students we are more certain of being at risk earlier on (*e.g.*, at the end of their first two terms) and wait for a few more terms before intervening with the ones we are less certain about.

4.5 Predicting Change of Major

As with identifying if a student is at risk of not graduating, there is a need of predicting whether a student will change majors both for the student's benefit as for the university's. Students are likely to waste significant time and energy by changing majors, and that could play a significant role in the time they take to graduate, if they end up graduating at all. With regards to departments, an important consideration when it comes to their funding is whether or not they can retain their students. If departments could more easily predict future losses in registrations, they could plan better and perhaps offer these students some guidance before they decide to change.

We define our problem as a binary supervised classification task: Predict whether students will change major at some point. For this task, we use the data described in section 4.2 (*i.e.*, a combination of demographics, student-level and institution-level features for students from Purdue University), with one modification. We remove all data entries that occur after they changed major, if at all they did. The reason for this change is that we want to predict changes of major before they change occurs and not after. If we are not able to predict it before they change, our efforts are not valuable. The labels we use for predicting change of major are whether the student

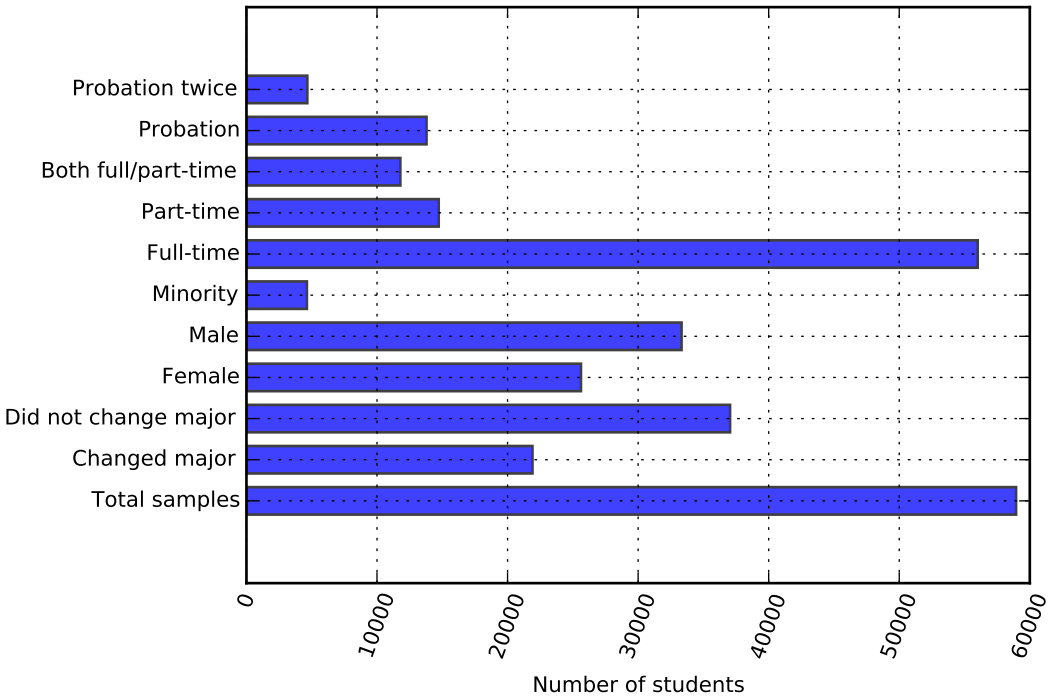

Figure 4.8.: Data exploration for predicting change of major

changed major at some point or not. That is, if they changed major at some point, the label is 1, and if they never changed major, the label is 0.

4.5.1 Features

Figure 4.8 provides us with some context before starting our modeling. It shows we have 58,947 student samples, out of which 21,906 of them changed major at some point during their academic program, while 37,041 of them never changed major. Out of the 58,947 students, 25616 identify as female and 33,331 identify as male. 4,598 students identify as belonging to an underrepresented group. 55,727 students were full-time students at some point during their studies, while 11,575 were part-time students at some point, and 9,189 were both full and part time students at some

Table 4.8.: Correlation between features and changes of major and respective p values

Feature	**Correlation to changing major**	p **value**
major change rate	0.461354036	< 0.0001
avg credits attempted	0.1412274903	< 0.0001
num terms	-0.134935702	< 0.0001
avg credits earned	0.1335675772	< 0.0001
full time	0.129397683	< 0.0001
avg credits gpa	0.1279155609	< 0.0001
part time	-0.0982282476	< 0.0001
gpa credits	-0.0825160128	< 0.0001
quality points	-0.0820466141	< 0.0001
probation once	-0.035404428	< 0.0001
female	-0.0304151572	< 0.0001
major grad rate	-0.0301776838	< 0.0001
gpa	0.0261676419	< 0.0001
probation twice	-0.0256149244	< 0.0001
avg credits failed	-0.0186220625	< 0.0001

points. Moreover, 9,189 students were at some point in academic probation, while 2,029 of them were in academic probation for at least two terms.

Table 4.8 shows correlations between the different features and changing major. From the data described in Section 4.2, we kept this set (presented in Table 4.8) of 15 features to use for our classification problem. According to this table, the most correlated feature is the change rate of the major the student is in. This is intuitive because if the student is in a major with a high change rate before changing, this student has a higher likelihood of changing than if the student is in a major with a low change rate. The second most correlated feature is the average number of credits the student has attempted per term, followed by the number of terms the student has been enrolled in before changing, which is negatively correlated.

It should be noted that all features are statistically significant with a p-value of less than 0.0001, for an $N = 58,947$. Some other features which did not prove statistically significant are included in the appendix.

4.5.2 Models

In order to predict whether a student is likely to change major, we use the data described in Section 4.2 and 4.5.1 and ran various models, as we did in Section 4.3 where we predicted whether students would withdraw (whether voluntarily or not) from their academic programs. We ran Logistic Regression, SVM, and Decision Tree models. All the features were used for the three models in order to determine which one is more suitable for this particular problem.

4.5.3 Experimental Results

To evaluate our models, we trained and tested them using the data described in Section 4.2 and with the features extracted described in Section 4.5.1. Our models were trained and tested with 5-fold cross validation. That is, with 80% of the data for training and the remaining 20% for testing. This means that we can use all the data for testing in 5 fold rotations.

Logistic Regression Modeling

Table 4.9 shows the coefficients of the logistic regression function trained with our data. The most contributing feature appears to be the last major change rate. This means that an increase in the change rate of the major will increase the likelihood of the student changing major. Similarly, being full time results in an increase in the likelihood of changing major. On the other side, an increase in the graduation rate of the last major the student, was in results in a decrease in the likelihood of changing major, as does being a part time student.

Table 4.10 shows the confusion matrix for the Logistic Regression model. Out of the 4,380 students in each rotation who changed major (1,833 + 2,547), the model was able to correctly identify 2,547 of them. This results in a recall of 58.2%.

Table 4.9.: Logistic regression model coefficients for predicting change of major

Feature	**Coefficient**
last major change rate	5.021798972
full time	0.8617504257
num terms	-0.2704430908
part time	0.2465644881
last major grad rate	0.2453293026
probation once	-0.107036565
female	0.0401022392
gpa	0.025806838
avg credits earned	0.0251575969
avg credits gpa	-0.0245922379
gpa credits	0.0172125186
avg credits attempted	0.0152636108
avg credits failed	-0.0098939861
probation twice	-0.0075770657
quality points	-0.0022937611

Table 4.10.: Logistic regression confusion matrix for predicting change of major

	Predicted stayed	Predicted changed
True stayed	5797	1611
True changed	1833	2547

Table 4.11.: SVM confusion matrix for predicting change of major

	Predicted stayed	Predicted changed
True stayed	6480	927
True changed	2818	1562

Support Vector Machines Modeling

Table 4.11 shows the confusion matrix for the SVM model. Out of 4,380 students in each rotation who changed major (2,818 + 1,562), the model was able to correctly identify 1,562 of them. This results in a recall of 35.7%.

Table 4.12.: Decision tree confusion matrix for predicting change of major

	Predicted stayed	Predicted changed
True stayed	5803	1604
True changed	1863	2517

Table 4.13.: Evaluation of models for predicting whether students will change major

	Precision	**Recall**	F_1 **score**	**Support**
Logistic regression	0.614	0.582	0.596	4381
Support vector machines	0.626	0.357	0.456	4381
Decision tree	0.612	0.575	0.592	4381

Decision Tree Modeling

Table 4.12 shows the confusion matrix for the decision tree model. Out of the 4,380 students in each rotation who changed major (1,863 + 2,517), the model was able to correctly identify 2,517 of them, which gives us a recall of 57.5%.

4.5.4 Model Comparison

Table 4.13 presents a comparison of the evaluation of our models. It shows the precision, recall, and F_1 measure for the different models built. SVM yields the highest precision of the three models, followed closely by logistic regression and decision tree. On the other hand, when we look at recall, we see that the logistic regression model performs the highest of the three, followed by the decision tree model, and then SVM with a significantly lower recall. This means that the logistic regression model was able to identify the highest number of students who would go on to change major.

Figures 4.9 and 4.10 show the performance results of each of the models. Predicting the likelihood of change of major did not prove as easy and straightforward as predicting withdrawal as we can see by the drop in the results shown in these figures.

All three models performed lower than in predicting withdrawal. Between models, logistic regression again performed better than decision trees and SVM. However, this

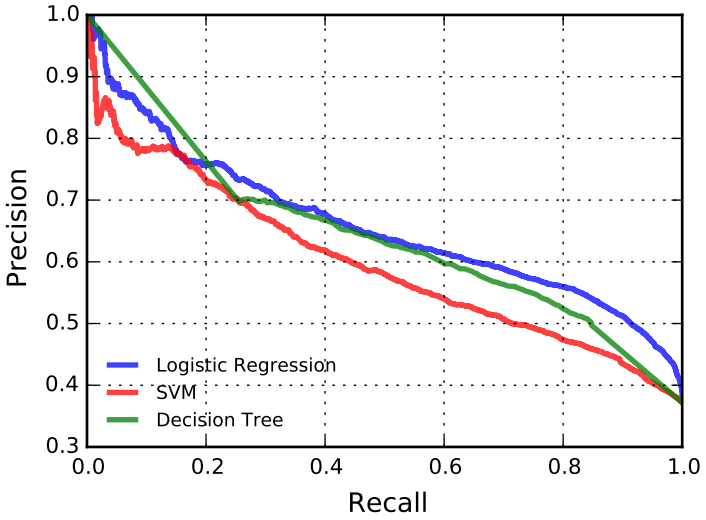

Figure 4.9.: Precision vs recall for each model in predicting student changes of major

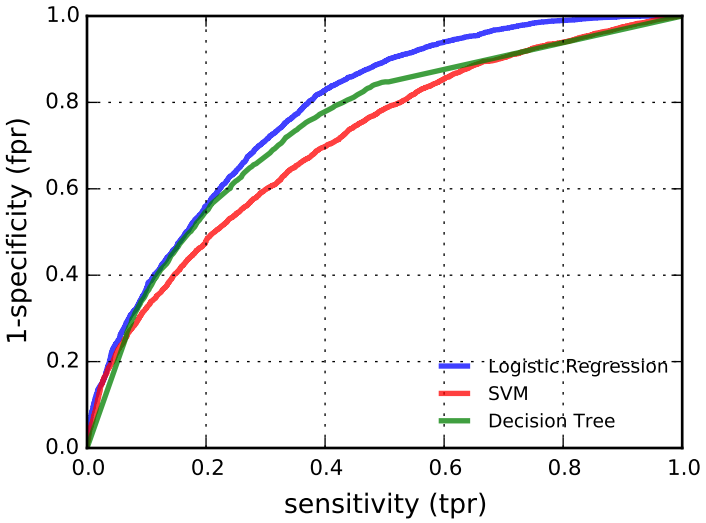

Figure 4.10.: ROC curve for each model for predicting student changes of major

time SVM performed significantly lower than the other two models. Using logistic regression, we can identify around 80% of the students who will go on to change major with an precision of 56%. Using decision trees, this number would be 52%, and 47% with SVM. If we wanted to identify 40% of the students who would change, we would

Table 4.14.: Evaluation of models: AUC values for various models for predicting change of major

Model	AUC
Logistic Regression	0.780
Decision Tree	0.742
Support Vector Machines	0.709

do so with a precision of 67% for logistic regression, 66% with decision trees, and 62% with SVM.

Table 4.14 shows the AUC scores for the three models. With these results we can again see that logistic regression outperforms the rest of the models, followed by the decision tree model.

4.5.5 Conclusions

In this section we proposed predicting whether a student will change major at some point during their academic program. We trained several machine learning models to classify students into two classes: change of major, and no change of major. As discussed in Section 4.5.4, we are able to identify 80% of students who would go on to change major with a precision of 67%. Although these results obtained are a little low, it is still an improvement over not using this modeling, as we are likely to help some students who could be struggling in the major they are in. Moreover, since we know the likelihood of each student changing major, we could set the threshold higher and only intervene with the students who are more likely to change (*i.e.* the ones we are more certain about). This approach further increases our precision and reduces type I error.

5 CONCLUSIONS AND FUTURE DIRECTIONS

5.1 Conclusions

In this dissertation we proposed various solutions to increase the effectiveness of educational technologies. In order for educational technologies to be effective they must not require a vast amount of time to maintain; they must not create unnecessary distractions; and they must be better at achieving academic goals than traditional educational methods, as discussed in [1]. Our contributions take these restrictions into account. In particular, we proposed the use of machine learning techniques to i) identify the most relevant and diverse discussions held in classrooms, in order to reduce distractions caused by irrelevant content and to reduce the time required for the instructor to manage the technology; ii) assist in tagging unlabeled content in intelligent tutoring systems so as to reduce the work required by teachers or content managers in the adoption of technology; and iii) predict students who are at risk of not graduating or of changing majors so as to be able to intervene early and provide them with the required guidance.

Firstly, we consider how online course discussions are usually carried out during class. We noticed that much of the content posted in such technologies is not relevant to the course content. Further, once we are able to identify these relevant questions or comments, we notice that many of these questions are repeated or very similar in nature. These are problems that cause these technologies to be less effective as irrelevant and repetitive content not only distract the students from the lecture content, but also require the instructor to go through much more content to be able to address the important questions. In Chapter 2 we proposed models to identifying the most relevant and diverse questions asked in online course discussions. We proposed two methods: a classification model that uses a combination of features from

the posts, lecture content, and similarity to old posts, in order to identify the most relevant content; and a ranking approach that uses submodularity and the similarity of the posts and lecture content in order to identify to most relevant and diverse questions. Both methods proposed proved to be effective at reducing the number of irrelevant and distracting content and at saving the instructor time in maintaining the technology. Particularly, in predicting the relevance of a post in an online course discussion, we found that the model that incorporates the topic modeling probability distribution together with the similarity (using KL divergence) of the post and the lecture, outperform the rest of the models studied. In ranking these discussion posts by relevancy and diversity of topics so as to over multiple aspects of the lecture, we found that using a combination of relevancy and submodularity outperform the rest of the models studied.

Secondly, we looked at how content creators set up intelligent tutoring systems. When creating new problems for the intelligent tutors, they need to select the knowledge components, or skills, associated to each problem. This is a tedious and time consuming task, especially when having to choose from dozens or hundreds of knowledge components. Further, not correctly choosing these knowledge components can cause the intelligent tutors to not work as effectively. In Chapter 3 we proposed models for suggesting knowledge components for untagged content in intelligent tutoring systems. These models are: text mining and an SVM classifier, and a query likelihood model with a kNN classifier. Both methods showed to be suitable for practical usage, thus increasing the functioning of the intelligent tutor's algorithms while also saving the content creator valuable time in setting up the technology. Our two models proposed consisted of a text mining approach with a multi-class SVM classifier, and a query likelihood search engine model with a kNN classifier. The SVM model outperforms the kNN model if the number of suggestions is low; however, if the number of suggestions increases, kNN tends to outperform the SVM model. In general, both approaches significantly improve the current way of selecting the associated knowledge components.

Thirdly, we consider how students, higher ed institutions, and even society, could benefit from the use of early warning systems that can identify students who are at risk of not graduating. Institutions invest much of their resources in identifying and admitting the students who are most likely to graduate, according to [59]. However, they many times do not invest as much of these resources in making sure they graduate. With institutional data from Purdue University, a large public university, we propose predicting student success by building models that would identify students who are likely to not graduate or to change majors, in Chapter 4. We use institutional data such as their registration records (the cohort they belong to, the department and major they are in, the classes they registered for), records that would give us a sense of progress (their GPA, their academic standing, whether they have been in probation), and demographics (self reported gender, whether they identify as belonging to a minority group). We modeled the data using three different classifiers: Support Vector Machines (SVM), Logistic Regression, and Decision trees. Our empirical results show we are able to make high precision predictions for identifying a large number of at-risk students. Furthermore, we studied how soon into their academic programs we can make high precision predictions in order to be able to intervene with these students as early as possible. Our results indicate that we can identify 80% of at-risk students with a precision of nearly 70% by the end of their third term at the university. If we waited a few more terms, *e.g.*, by the end of their fourth term, we could achieve a precision of 73% for identifying 80% of at-risk students. Moreover, our empirical results for identifying students who would go on to change majors show that we can identify 80% of these students with a precision of 56%. If we aimed at identifying 40% of the students who would go on to change majors, we would do so with a precision of 67%.

5.2 Future Directions

This dissertation proposed novel methods for increasing the effectiveness of educational technologies. However, there is still much room for extending this work. This section discusses some possible directions that would be beneficial for the future of educational technologies.

5.2.1 Online Course Discussions

A possible direction to consider when it comes to increasing the effectiveness of technologies that offer online course discussions is to model knowledge transfer from one student to another by analyzing questions and responses (which our study did not consider). In addition, it would be valuable to include the input of the instructor in a qualitative analysis. Furthermore, this study could be extended to more courses in different domains to ensure its validity in more contexts.

5.2.2 Intelligent Tutoring Systems

The use of semantic features should be explored. For example, including extracted from topic modeling and the presence of part of speech terms. With the use of topic modeling features, perhaps we could identify some topics which are more indicative of certain KCs. Likewise, we might be able to associate the use of certain part-of-speech tags with specific KCs. Further, it would be beneficial to this study to explore different domains.

5.2.3 Early Warning Systems

Our work would likely benefit from integrating students' data from CMS academic progress. This would provide us with a lower level of academic progress than what we used in our study, in which we only had access to final course grades. Using CMS data we might be able to identify at-risk students within the semester and intervene

even earlier. Another likely improvement would come from integrating students' perceptions on how well they are doing in their studies. Purdue University recently developed a technology aimed at identifying study patterns where students can keep track of their study habits and rate their perceived productivity. Including student perception features into our models would be an interesting direction to explore. This approach might identify students who are at-risk and believe they are not, which would be valuable information for their mentors to have when intervening with them.

REFERENCES

REFERENCES

[1] Yong Zhao and Gary A Cziko. Teacher adoption of technology: A perceptual control theory perspective. *Journal of Technology and Teacher Education*, 9(1):5–30, 2001.

[2] Thomas A Beggs. *Influences and Barriers to the Adoption of Instructional Technology*. State University of West Georgia, Learning Resources, 2000. Technical Report.

[3] George Zhou and Judy Xu. Adoption of educational technology ten years after setting strategic goals: A Canadian university case. *Australasian Journal of Educational Technology*, 23(4), 2007.

[4] Studio by Purdue, Purdue University. Hotseat. `www.openhotseat.org`. Accessed: 2015.

[5] Kurt VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011.

[6] Pittsburgh Science for Learning Center. Learnlab. `http://www.learnlab.org/research/wiki/index.php/Knowledge_component`. Accessed: 2011.

[7] Kenneth R. Koedinger, Albert T. Corbett, and Charles Perfetti. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, 36(5):757–798, 2012.

[8] Kerstin Borau, Carsten Ullrich, Jinjin Feng, and Ruimin Shen. Microblogging for language learning: Using Twitter to train communicative and cultural competence. In *Proceedings of the International Conference on Advances in Web Based Learning (ICWL)*, pages 78–87, 2009.

[9] Cristina Costa, Guenter Beham, Wolfgang Reinhardt, and Martin Sillaots. Microblogging in technology enhanced learning: A use-case inspection of PPE summer school 2008. In *Proceedings of the Second Workshop on Social Information Retrieval for Technology Enhanced Learning (SIRTEL)*, 2008.

[10] Gabriela Grosseck and Carmen Holotescu. Can we use Twitter for educational activities? In *Proceedings of the Fourth International Scientific Conference on eLearning and Software for Education (eLSE)*, 2008.

[11] Carsten Ullrich, Kerstin Borau, Heng Luo, Xiaohong Tan, Liping Shen, and Ruimin Shen. Why web 2.0 is good for learning and for research: Principles and prototypes. In *Proceedings of the International Conference on World Wide Web (WWW)*, pages 705–714. ACM, 2008.

[12] Meredith Ringel Morris, Scott Counts, Asta Roseway, Aaron Hoff, and Julia Schwarz. Tweeting is believing?: Understanding microblog credibility perceptions. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW)*, pages 441–450. ACM, 2012.

[13] Suleyman Cetintas, Luo Si, Sugato Chakravarty, Hans Aagard, and Kyle Bowen. Learning to identify students' relevant and irrelevant questions in a microblogging supported classroom. In *Proceedings of the International Conference on Intelligent Tutoring Systems (ITS)*, pages 281–284, 2010.

[14] Suleyman Cetintas, Luo Si, Hans Aagard, Kyle Bowen, and Mariheida Cordova-Sanchez. Microblogging in a classroom: Classifying students' relevant and irrelevant questions in a microblogging-supported classroom. *Proceedings of the IEEE Transactions on Learning Technologies (TLT)*, 4(4):292–300, 2011.

[15] Huizhong Duan, Yunbo Cao, Chin yew Lin, and Yong Yu. Searching questions by identifying question topic and question focus. In *Proceedings of 46th Annual Meeting of the Association for Computational Linguistics: Human Language Tchnologies (ACL:HLT)*, 2008.

[16] Peng Han, Ruimin Shen, Fan Yang, and Qiang Yang. The application of case based reasoning on q&a system. In *Proceedings of the 15th Australian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence (AI)*, pages 704–713, 2002.

[17] Jiwoon Jeon, W. Bruce Croft, and Joon Ho Lee. Finding semantically similar questions based on their answers. In *Proceedings of the International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 617–618. ACM, 2005.

[18] Valentin Jijkoun. Retrieving answers from frequently asked questions pages on the web. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*. ACM, 2005.

[19] Kai Wang, Zhaoyan Ming, and Tat-Seng Chua. A syntactic tree matching approach to finding similar questions in community-based qa services. In *Proceedings of the International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 187–194. ACM, 2009.

[20] Maggy Anastasia Suryanto, Ee Peng Lim, Aixin Sun, and Roger H. L. Chiang. Quality-aware collaborative question answering: Methods and evaluation. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*, pages 142–151. ACM, 2009.

[21] Deepak P. and Sutanu Chakraborti. Finding relevant tweets. In *Proceedings of the 13th International Conference on Web-Age Information Management (WAIM)*, pages 228–240, 2012.

[22] Young-In Song, Chin-Yew Lin, Yunbo Cao, and Hae-Chang Rim. Question utility: A novel static ranking of question search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2008.

[23] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[24] Mariheida Cordova-Sanchez, Parameswaran Raman, Luo Si, and Jason Fish. Relevancy prediction of micro-blog questions in an educational setting. In *Proceedings of the International Conference on Educational Data Mining (EDM)*, pages 415–416, 2014.

[25] Daniel Ramage, Susan T Dumais, and Daniel J Liebling. Characterizing microblogs with topic models. *Fourth International Conference on Weblogs and Social Media (ICWSM)*, 10:130–137, 2010.

[26] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 137–146. ACM, 2003.

[27] Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT) - Volume 1*, pages 510–520. Association for Computational Linguistics, 2011.

[28] Amr Ahmed, Choon Hui Teo, S.V.N. Vishwanathan, and Alex Smola. Fair and balanced: Learning to present news stories. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*, pages 333–342. ACM, 2012.

[29] Khalid El-Arini, Gaurav Veda, Dafna Shahaf, and Carlos Guestrin. Turning down the noise in the blogosphere. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 289–298. ACM, 2009.

[30] Karthik Raman, Pannaga Shivaswamy, and Thorsten Joachims. Online learning to diversify from implicit feedback. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 705–713. ACM, 2012.

[31] Matthew D. Hoffman, David M. Blei, and Francis Bach. Online learning for latent dirichlet allocation. In *Proceedings of the International Conference on Neural Information Processing Systems (NIPS)*, pages 856–864, 2010.

[32] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, 2010.

[33] Mark Steyvers, Padhraic Smyth, Michal Rosen-Zvi, and Thomas Griffiths. Probabilistic author-topic models for information discovery. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 306–315. ACM, 2004.

[34] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing Twitter and traditional media using topic models. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval (ECIR)*, pages 338–349, 2011.

[35] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP), Volume 1*, pages 248–256. Association for Computational Linguistics, 2009.

[36] Satoru Iwata, Lisa Fleischer, and Satoru Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM (JACM)*, 48(4):761–777, 2001.

[37] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978.

[38] Mariheida Cordova-Sanchez and Pinar Yanardag. Turning down the noise in classrooms. In *Proceedings of the International Conference on World Wide Web (WWW), Companion Volume*, pages 905–910, 2016.

[39] RSJD Baker. Data mining for education. *International Encyclopedia of Education*, 7(3):112–118, 2010.

[40] Thorsten Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning (ECML)*, pages 137–142, 1998.

[41] University of Massachusetts and Carnegie Mellon University. The Lemur Project. `www.lemurproject.org`. Accessed: 2013.

[42] Worcester Polytechnic Institute. ASSISTments. `https://www.assistments.org/`. Accessed: 2011.

[43] Leena Razzaq, Jozsef Patvarczki, Shane F. Almeida, Manasi Vartak, Mingyu Feng, Neil T. Heffernan, and Kenneth R. Koedinger. The assistment builder: Supporting the life cycle of tutoring system content creation. *Proceedings of the IEEE Transactions on Learning Technologies (TLT)*, 2(2):157 – 166, 2009.

[44] Abigail S. Gertner and Kurt VanLehn. Andes: A coached problem solving environment for physics. In *Proceedings of the International Conference on Intelligent Tutoring Systems (ITS)*, pages 133–142, 2000.

[45] Kenneth R. Koedinger, John R. Anderson, William H. Hadley, and Mary A. Mark. Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education (IJAIED)*, 8:30–43, 1997.

[46] Albert T. Corbett and John R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1994.

[47] Carolyn Rosé, Pinar Donmez, Gahgene Gweon, Andrea Knight, Brian Junker, William Cohen, Kenneth Koedinger, and Neil Heffernan. Automatic and semi-automatic skill coding with a view towards supporting on-line assessment. In *Proceedings of the 2005 Conference on Artificial Intelligence in Education: Supporting Learning Through Intelligent and Socially Informed Technology*, pages 571–578, 2005.

[48] Menucha Birenbaum, Anthony E. Kelly, and Kikumi K. Tatsuoka. Diagnosing knowledge states in algebra using the rule space model. *Proceedings of the Educational Testing Services (ETS) Research Report Series*, 1992(2):i–25, 1992.

[49] Hao Cen, Kenneth Koedinger, and Brian Junker. Learning factors analysis – a general method for cognitive model evaluation and improvement. In *Proceedings of the International Conference on Intelligent Tutoring Systems (ITS)*, pages 164–175, 2006.

[50] Michel Desmarais. Conditions for effectively deriving a q-matrix from data with non-negative matrix factorization. best paper award. In *Proceedings of the International Conference on Educational Data Mining (EDM)*, pages 41–50, 2011.

[51] Leena Razzaq, Neil T Heffernan, Mingyu Feng, and Zachary A Pardos. Developing fine-grained transfer models in the assistment system. *Journal of Technology, Instruction, Cognition, and Learning*, 5(3):289–304, 2007.

[52] Institute Jozef Stefan Artificial Intelligence Laboratory. Text Garden. `http://ailab.ijs.si/tools/text-garden/`. Accessed: 2011.

[53] Suleyman Cetintas, Luo Si, Yan Ping Xin, and Dake Zhang. Automatic text categorization of mathematical word problems. In *Proceedings of the 22nd International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, pages 27–32, 2009.

[54] Mario Karlovcec, Mariheida Cordova-Sanchez, and Zachary A. Pardos. Knowledge component suggestion for untagged content in an intelligent tutoring system. In *Proceedings of the International Conference on Intelligent Tutoring Systems (ITS)*, pages 195–200, 2012.

[55] Kenneth C. Green. The 2005 campus computing survey. In *The Campus Computing Project*, 2005. Technical Report.

[56] Philip Goldstein, Richard Katz, and EDUCAUSE Center for Applied Research. *Academic Analytics: The Uses of Management Information and Technology in Higher Education*. Research study from the EDUCAUSE Center for Applied Research. 2005.

[57] Vincent Tinto. Dropout from higher education: A theoretical synthesis of recent research. *Review of Educational Research*, 45(1):89–125, 1975.

[58] A.W. Astin. *What Matters in College: Four Critical Years Revisited*. Jossey-Bass Higher and Adult Education Series. 1997.

[59] J.P. Campbell. *Utilizing Student Data Within the Course Management System to Determine Undergraduate Student Academic Success: An Exploratory Study*. Purdue University, 2007. PhD Dissertation.

[60] Kimberly E Arnold. Signals: Applying academic analytics. *Educause Quarterly*, 33(1):n1, 2010.

[61] Kimberly E Arnold and Matthew D Pistilli. Course signals at Purdue: Using learning analytics to increase student success. In *Proceedings of the Second International Conference on Learning Analytics and Knowledge (LAK)*, pages 267–270. ACM, 2012.

[62] Kimberly E Arnold, Zeynep Tanes, and Abigail Selzer King. Administrative perceptions of data-mining software signals: Promoting student success and retention. *The Journal of Academic Administration in Higher Education*, 6(2):29–39, 2010.

[63] Jeffrey Sklar. *The Impact of Change of Major on Time to Bachelors Degree Completion with Special Emphasis on STEM Disciplines: A Multilevel Discrete-Time Hazard Modeling Approach*. California Polytechnic State University, 2014. Final Report.

[64] Clinton I. Chase and John M. Keene. Major declaration and academic motivation. *Journal for College Student Personnel*, 22(6):496–502, 1981.

[65] Kelsey Dietz. *Predicting Undergraduate Passions: An Analysis of Major Migration at Carnegie Mellon*. Carnegie Mellon University, 2015. Thesis.

APPENDIX

APPENDIX

Table A.1.: Correlation between features and the student withdrawing (voluntarily or not) and respective p values, with an N of 37,162

Feature	Correlation to withdrawing	p value
quality points	-0.624234943135	< 0.0001
last major graduation rate	-0.538978426731	< 0.0001
gpa	-0.523590979745	< 0.0001
gpa credits	-0.504342353891	< 0.0001
last college graduation rate	-0.466680645281	< 0.0001
avg credits earned	-0.455278198202	< 0.0001
avg credits failed	0.453064186389	< 0.0001
probation	0.399578299083	< 0.0001
last college graduation rate	-0.388570586012	< 0.0001
number of terms	-0.356210852028	< 0.0001
avg credits gpa	-0.275950599814	< 0.0001
probation twice	0.222828163194	< 0.0001
full time	-0.223098341242	< 0.0001
avg credits attempted	-0.21018610131	< 0.0001
minority	0.0797970949683	< 0.0001
changing major	-0.074641024616	< 0.0001
female	-0.0419017460189	< 0.0001
change college higher grad rate	-0.0347672960694	< 0.0001
change college	-0.0339250512379	< 0.0001
part time	0.0189895734924	< 0.0002
change college lower grad rate	-0.00738480144865	0.15

Table A.2.: Correlation between features and the student changing major and respective p values, with an N of 58,947

Feature	Correlation to withdrawing	p value
major change rate	0.461354036	< 0.0001
avg credits attempted	0.1412274903	< 0.0001
num terms	-0.134935702	< 0.0001
avg credits earned	0.1335675772	< 0.0001
full time	0.129397683	< 0.0001
avg credits gpa	0.1279155609	< 0.0001
part time	-0.0982282476	< 0.0001
gpa credits	-0.0825160128	< 0.0001
quality points	-0.0820466141	< 0.0001
probation once	-0.035404428	< 0.0001
female	-0.0304151572	< 0.0001
major grad rate	-0.0301776838	< 0.0001
gpa	0.0261676419	< 0.0001
probation twice	-0.0256149244	< 0.0001
avg credits failed	-0.0186220625	< 0.0001
last major avg gpa	0.010959125	< 0.0078
minority	0.0020005274	0.627

VITA

VITA

Mariheida Córdova Sánchez was born and raised in Puerto Rico. She joined the University of Puerto Rico, Mayagüez, for her BS in computer engineering. She spent a summer doing research at the Virginia Polytechnic Institute and State University, another summer at the University of California, Berkeley, and another as an intern at Intel.

After graduating from UPR Mayagüez, Mariheida joined Purdue University for her Ph.D. in computer science. At Purdue, she had the opportunity of working on research problems in the areas of information retrieval, machine learning, and education. While working toward her Ph.D. degree, she spent time working for Purdue's Teaching and Learning Technologies team in applying machine learning to educational technologies. Along the way, she received a MS degree in computer science as well. Mariheida has been involved in many outreach programs and recruiting events, particularly for minorities, and served on the CS Graduate Student Board, and as the President, Secretary, and Treasurer of the Puerto Rican Student Association at Purdue.