Analysis of Student Data for Retention Using Data Mining Techniques

Brandon Sherrill
Dept. of Computer Science
Tennessee Technological University
blsherrill42@students.tntech.edu

William Eberle
Dept. of Computer Science
Tennessee Technological University
weberle@tntech.edu

Doug Talbert
Dept. of Computer Science
Tennessee Technological University
dtalbert@tntech.edu

Abstract - At most universities, administrators and counselors are trying to devise sound methodologies to help increase student retention rates. Due to the vast amount of student data that is available, sorting through the data to extract useful knowledge is a daunting task. However, the data may be helpful in predicting future student trends – particularly as it relates to retention. In this paper, we describe data mining and machine learning techniques that can be used to predict future enrollment. In our experiments, we attempt to apply these techniques to the retention of Computer Science students – a major that traditionally has significant turnover in the first year of study. Specific algorithms are selected to classify the data in an attempt to extract relevant information. While traditional methods may focus on known issues with retention, we emphasize the importance of factors that may only be noticeable through the application of data mining. The goal of this research is to determine which newly enrolled students should be targeted as retention risks. If these students can be identified early in their academic career, this approach may help increase retention rates not only within a specific department, but across the entire university.

Introduction

The work described in this paper analyzes the student retention problem through data mining by attempting to analyze all student data that is readily available to our department, the Department of Computer Science at Tennessee Technological University (TTU). In addition, the data records used in the following experiments are available to most departments at colleges and universities, which will allow us to generalize our methodologies to a wide variety of retention cases.

The study of this student data brings together two seemingly unrelated fields: student retention and data mining. Student retention is a problem that is traditionally investigated using statistical approaches. In this work, the use of data mining (and also machine learning) techniques help us better understand which attributes contribute to the student retention problem.

Student Retention

The problem of student retention is one that higher-education departments are interested in because of the impact on enrollment. The long term goal of this project is to reduce the negative impact that poor student retention has on our department and university. Traditionally, student retention is investigated using statistical approaches, such as logistic regression (Chang 2006). While statistical approaches work adequately in some cases, we will demonstrate that various data mining/machine learning techniques provide a greater amount of testing customization and flexibility.

Data Mining Applied to Retention

The use of data mining for student retention can result in novel discoveries that traditionally cannot be observed with the more statistical approaches. In order to apply our machine learning techniques (which will be discussed in a subsequent section), we followed several data mining steps, which can then

be generalized for a wider applicability. In this process, the department must first obtain student data from the university. Once that data is acquired, it must be cleansed. The cleansing of the data involves formatting of the data so that it is free from any irregular information. Once the data is properly formatted (and appropriate data types chosen), it can be inserted into the database designated for the retention analysis project. It is this database that provides the structure of the data to be tested, and this structure is crucial to a better understanding of the data. Once the database is set up, the cleansed data can be inserted into the database. When the data is placed into the database, a myriad of combinations of elements can be considered for testing purposes. Once all elements needed for a particular test set are determined, the data may be exported in a variety of formats. A file format must be selected, and then the data elements can be transferred to the file. This file can then be analyzed by the machine learning algorithms. It is through the outputs of these algorithms that observations can be made about the retention data.

Related Works in Educational Data Mining

Since this paper is focused on data mining and its application to student retention, any works described here are done so in order to emphasize the use of data mining in the educational field. Special attention has been given to research that involves specific student retention data mining studies.

One of the first detailed educational applications of knowledge discovery was a study conducted by Sanjeev and Zytkow (1995). They discuss an analytical approach that uses first-time freshmen as the entries in their data set. This dataset's attributes are then divided into three categories: demographics, high school performance, and university performance. Sanjeev and Zytkow conclude that the high school academic results, such as GPA, are the best predictors of college performance. In fact, it is found that high school GPA is a better predictor of college performance than either class rank or ACT composite scores (Sanjeev and Zytkow 1995).

The first major work to give an objective summary of data mining's application to education was Luan (2002). Luan presents a general summary of data mining to the educational field and then follows with a case study of his own (Luan 2002). While several data mining approaches have been used in testing student academic performance since then, only a few studies have been completed in regard to student retention.

Barker et al. (2004) is one of the first data-mining-based retention studies and has some similarities to our work. Their work notes that a student's classification is important to determine as soon as possible. In a study that involves incoming freshmen student data from the fall of 1995, 1996, and 1997, Barker et al. set out to classify these students into two groups: students who receive an undergraduate degree within six years and students who do not. The student data includes an incoming freshmen survey, demographic data, and high school academic data. It should be noted that any student that did not contain all data elements was excluded from their tests. The data sets were grouped for analysis in three different ways: a large, combined test set, a test set between years, and test set within years. This study used artificial neural networks and support vector machines for student classification. In general, accuracies with these algorithms ranged from sixty to seventy percent. The study concludes that the largest test set that spanned all three years gave the most generalized results (Barker et al. 2004). However, their work mostly involved student surveys with very specific questions, and as such, is difficult to generalize to a wider range of institutions.

Zhang and Clark (2010) conducted another retention study using data mining that should be included in this list. Like Barker et al., this study focused on the importance of early student classification. The importance of student intervention is also noticed, as it necessary once an at-risk student has been identified. This paper is unique, as it proposes a complete knowledge discovery system, referred to as the Mining Course Management System. This system is supposed to encompass all data mining systems and allow for a large amount of data integration. The experimental portion of this paper uses only one year of student data for testing, which could lead to a very specified training model. The algorithms used for

student retention testing are naïve Bayes, support vector machine, and decision trees. The classification accuracy ranges from eighty to ninety percent, but these accuracies may not be reliable, since only a year of data is being tested. In this study, specific algorithm parameters are not mentioned, and the logic used to determine retention status is not given (Zhang and Clark, 2010).

Dekker (2009) is the most recent data-mining-based retention study that is most similar to our own. The structure of the Dutch education system described in the study differs slightly from our case, but the material is very relevant. This thorough retention study describes a department whose current retention methods are based loosely on past student similarities, which is similar to the retention situation at our university. This study, like ours, makes an effort to focus on data that is readily available to the university. As the previous two studies have stressed, this paper also focuses on the early detection of at-risk students. This study uses the Weka suite of data mining/machine learning algorithms (Hall et al. 2009). The data set of this study includes all bachelors' students in the electrical engineering department that are first year freshmen. Once the data has been filtered, Dekker divides the dataset into past education and university grades. The classification system of the study divides students into three groups (bad, at-risk, good) based on their credits completed within a year. It is interesting to note that Dekker's retention classification is determined after only a single year, which could cause misleading and inaccurate results. Through the use of decision tree classifiers, Bayesian classifiers, logistic models, rule-based learners, and random forests, this study manages to get an accuracy percentage between seventy five and eighty percent using all combined data. However, their study concludes that the best information for testing purposes is *university data* instead of pre-university data. Dekker shows that decent results can be obtained when using basic university data without having to collect external student data.

It should also be noted that other studies have examined similar topics. For instance, Yu offers a study of retention beyond the first year (Yu 2010). Also, a study by Herzog focuses on estimating student retention along with the students' total time until degree completion (Herzog 2006). As more studies such as these are completed, the attributes of the datasets, the logic that defines student retention, and the urgency of student classification become crucial for consistent comparisons of studies.

Student Retention Data

Our hypothesis is that data mining techniques must be combined with machine learning algorithms so that an accurate retention classification can be made as early as possible for each student. In addition, the accuracy of this classification will be influenced by various aspects in the data, which means that the data must be free from any major errors.

Initial Steps

Before any major work can occur, the student data must be acquired from a university source. At TTU, there is not a centralized server that contains all student data in one place. Instead, this data is kept on various servers around campus. One can imagine that this same scenario might exist across many campuses. Our data existed as two comma-separated value (csv) files upon arrival. One file contained 1094 entries related to high school student information, and the other contained 1152 entries that related to university term data. This data represented computer science students from the fall of 2003 to the spring of 2007. It must be noted that Weka was used as the sole software package for this project. In these initial steps, Weka was used to open the csv files and explore the makeup of the data. On initial inspection, both files contained oddities in the data such as missing values, improperly formatted entries, and several entries that stored one data field in another's place. Needless to say, the data has to be cleansed of all irregularities before any progress can be made. Once the data is free from noticeable errors, it must be properly formatted and then inserted into a central project database, known as a data warehouse (Han and Kamber 2006). This database is a location where only the cleansed data is stored and it will become the starting point for all future tests.

Data Warehouse

The data warehouse is a database where all relevant student data is stored. The structure of our database is shown in Figure 1. There are seven tables in the database, each of which contains different elements that are loosely related to each other. The two files mentioned earlier take up three tables: *Students*, *Student_Terms*, and *High_Schools*. The remaining four tables are currently being used to incorporate student course information, which is described in the future work section.

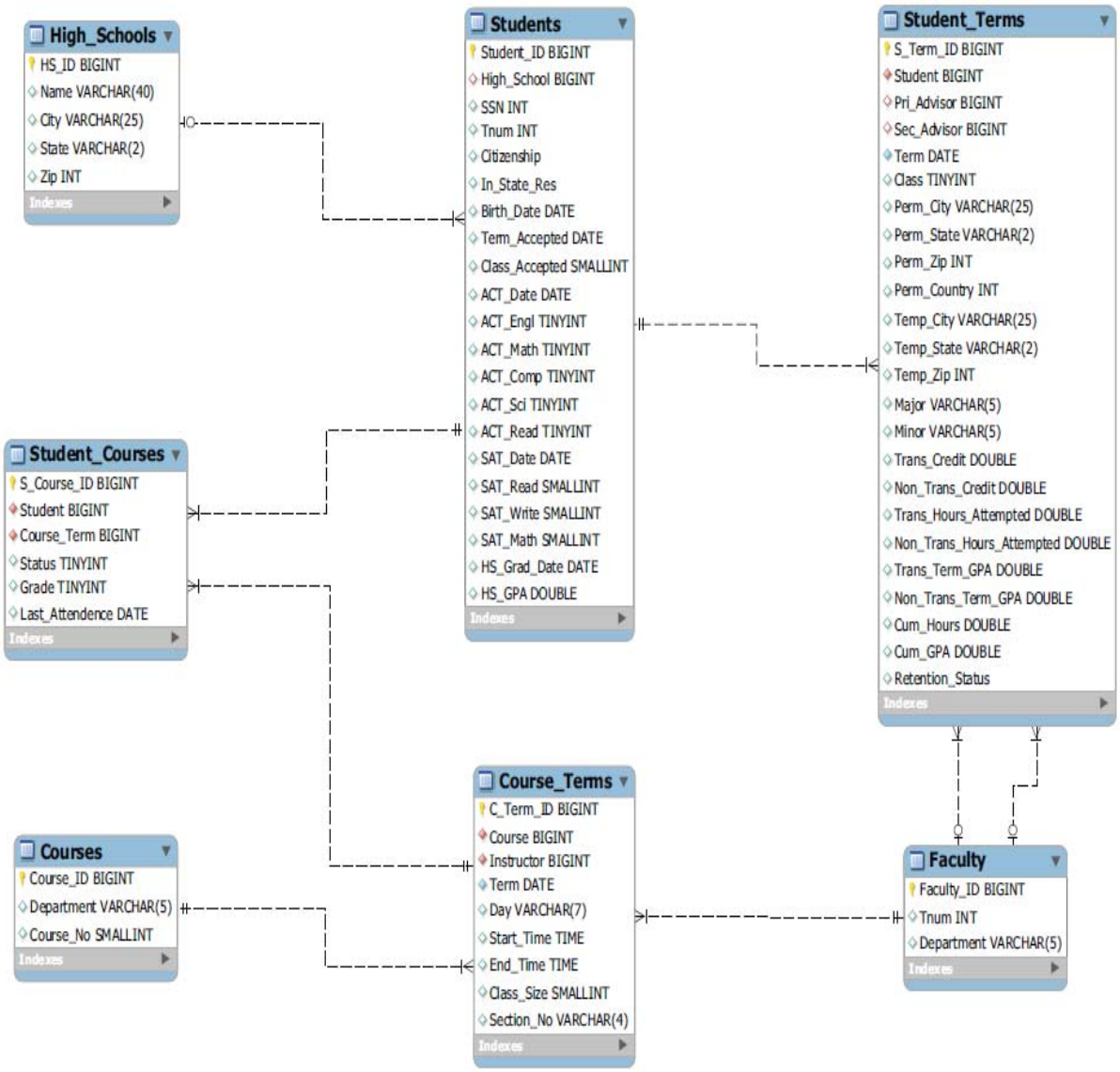

Figure 1: Data Warehouse Schema

As shown in Figure 1, one will observe that the data elements used in the tables are not unique to our department. In fact, these elements are not specific to any department, which means that this same schema could most likely be used within other departments at TTU. Also, the information contained in the database is not specific to the university or location in the country. In addition, the simplicity of the schema allows for a generalization to other universities.

By organizing the data into this data warehouse, we can then analyze the data from multiple directions. In our tests, all three populated tables were exported from the database in their entirety. These tables were inserted into a csv file by row, each row containing a student's particular term information,

personal information, and high school information. Once in the file, any unnecessary columns, such as database identification numbers and student-specific ID numbers, are removed. Zip codes were also removed, as one numeric value may represent a diverse geographic area (which could hinder the classification process).

Preprocessing

When the data is removed from the database and placed in a file, it is then capable of being loaded into Weka for preprocessing steps. As discussed earlier, while the data was initially run through the Weka tool, the data had not been cleansed or analyzed yet. Now that it is cleansed, we can actually make some optimization changes to the data so that it is better prepared for classification. Some of the numeric attributes, such as test scores, GPAs, and hours, are grouped into bins using the *Discretize* function of Weka. This allows the classification algorithms to see a smaller set of data, which makes classification less complex. All other numeric values are converted to nominal values using the *NumericToNominal* function. This conversion permits the non-numeric representation of numeric values in the form of unique identifiers. By using these identifiers, the classification algorithms can keep all of the values distinguished. It must be mentioned that, in general, the classification algorithms did not seem to have problems coping with missing values, which is why no effort was put into developing a solution for missing values.

Classification

Up until this point, we have not defined *student retention*. It is important to understand the logic that each data mining study uses to define retention, as it has an effect on the results. A degree from the department of computer science typically takes 120 credit hours to complete, so we declare a student retained once they have completed 90 hours. While this student may drop out before they receive a degree, we consider the student's decision to be unrelated to the university. In this study, only two classes are used to represent students: retained or not retained.

The role of classification algorithms is to read in all data given to them and construct a model based on the algorithm's training of the data. Once this model has been produced, part of the data is tested against the model to estimate the accuracy of the original model. The accuracy of a classifier is determined by how well it groups, or classifies, the data into the two classes. Table 1 shows a confusion matrix that describes the classification used in this study. A perfect classification, one hundred percent accuracy, is achieved when all classifications fall under the true positive and true negative fields. Any instance of a false positive or false negative introduces an inaccuracy into the classification. This occurs when the predicted class does not match up with the actual class. While neither false positives nor false negatives are desired, false positives are worse for our study, since, in that case, a student is predicted to be retained, when they are actually not. Clearly, it would be better to mislabel more students as being at risk and intervene unnecessarily, rather than missing them completely.

Confusion Matrix		**Classified**	
		Retained (Not At-Risk)	Non-retained (At-risk)
Actual	Retained	*True Positive*	*False Negative*
	Non-retained	*False Positive*	*True Negative*

Table 1: Retention Confusion Matrix

The five classifiers chosen for this work are as follows.

Decision Tree

A decision tree classifier creates a tree structure based on the elements within the data. Attributes are set as nodes and the classification (retained or non-retained) are set as the leaves of the nodes. Once the tree is created with the training data, the test data is sent through the tree and the predictions are recorded and used to calculate the accuracies. Decision tree algorithms use attribute selection features to determine how data is to be split at each node, and tree pruning is used to remove any tree branches that are considered unreliable. The tree structure of the model that is created makes the decisions easy to understand and allows rules to be extracted easily (Han and Kamber 2006). The J48 (C4.5) algorithm was used during testing in Weka (Quinlan 1993). The default settings for a pruned tree were selected.

Bayesian Classifier

A Bayesian classifier is a statistical classifier that predicts a class based on probabilities. This prediction is based on Bayes theorem (Mitchell 1997). The naïve Bayesian algorithm treats all data elements as independent of one another, which is known as class conditional independence (John & Langley 1995). The Naïve Bayes algorithm was chosen because of its documented speed for classifying (Hans and Kamber 2006).

Neural Net

The neural net classifier is based upon a simplistic model of human neurons, and the classification model is similar in structure. The network is made up of an input layer, a hidden layer(s), and an output layer. The input layer is where all data elements are fed into the algorithm, and the output layer is where the predictions come out. It is the hidden layer(s) where the complexity lies. Backpropogation occurs in the hidden layers using a weighting scheme (Hans and Kamber 2006). Using the Multilayer Perceptron algorithm provided by Weka, four parameters (learning rate (0.1), momentum (0.1), hidden layers (5), and training time (500)) were adjusted to reduce the running time for this approach (Mitchell 1997). Various combinations of these parameters were considered, and the values listed above provided optimal accuracy while maintaining a decreased runtime.

Support Vector Machine

An SVM is a classifier that maps two-class data to a multi-dimensional space. The algorithm attempts to separate the two distinct classes in space with a dividing line, known as a hyperplane. Using the hyperplane and margin lines, the largest distance between the two nearest data elements, or support vectors, is found. This distance is known as the maximum marginal hyperplane, and is used for determining the best classification (Hans and Kamber 2006). The SMO SVM algorithm was used with Weka, and the parameters were left at their default settings (Platt 1998).

Lazy Learner

A lazy learner, unlike the algorithms already described, does not build a model until test data is sent to it. These classifiers do less work during the training phase and more work during testing. The Ibk algorithm was used in this study, and it is an example of a k-nearest neighbor algorithm (Aha & Kibler 1991). Lazy learners compare a given test data element with similar previously classified elements that they find. These similar elements are known as nearest neighbors, and the number of them can be specified during testing. In our testing, we left that number as a default. Once these neighbors are found, the new test element is given the class of the most common class among the nearest neighbors.

Training and Testing

Once the data has been processed for classification, one of the classification algorithms must be selected and the parameters must be set. In our study, all data consisted of known retention values, meaning that there were no students who had an unknown retention status. Therefore part of the data

must be used for training, and the rest must be used for testing. In Weka, we chose to use ten-fold cross validation testing, which partitions the data into ten equally-sized pieces. Ten tests are run, each using a different piece of the data as test data and the remaining nine pieces as training data. Once all ten tests are run, the results are averaged to get the final output. While the algorithms above have their specific outputs, all algorithm tests share a common accuracy measurement – the confusion matrix described earlier. This allows for a performance comparison between all of the algorithms. Algorithm execution times are another way to compare the performance of the different algorithms. However, each of the five algorithms took only a few seconds to complete, so the runtimes were disregarded as viable comparison information.

In our initial tests, students whose retention status could be determined were grouped by the number of terms that they were enrolled in the department. *Term1* students were in the department for at least one term, *Term2* for at least two terms, *Term3* for at least three terms, and *Term4* for at least four terms. Other than this separation, all students within each term are grouped as one body. Each test set included information taken from the *Students*, *High_Schools*, and *Student_Terms* tables. See Table 2 for the comparison of algorithm accuracies of these tests.

	Term1	Term2	Term3	Term4
Ibk	73.3	68.6	67.8	74.6
J48	81.7	80.5	81.6	88.9
Multilayer Perceptron	62.3	55	80.5	85.7
NaiveBayes	79.9	73.4	67.8	46
SMO	82.8	81.1	81.6	85.7

Table 2: Initial Four Term Classification Accuracies

During testing, several methods were used to attempt to increase accuracy, such as bagging, boosting, and attribute selection, but no noticeable accuracy gain was found with these algorithms and test sets. After the results shown in Table 2, we determined that one of the most important test sets was not included - pre-university student data (Term0). This decision goes against the findings of Dekker (2009), who determined that university data was the best available. This set includes the *Students* and *High_Schools* tables, but not the *Student_Terms* table. Also, we decided that, along with the pre-university test set, only the first term (*Term1*) test set should be used. If decent accuracies could be obtained with these two test sets, students can be classified after their first semester in college. Another change that was needed for this data was a separation of transfer students and non-transfer students. We felt that this distinction would help the algorithms make a better classification, and that was the case. See Table 3 for a table containing accuracies of the two test sets split into transfer and non-transfer students.

	Term0 Nontransfer	Term0 Transfer	Term1 NonTransfer	Term1 Transfer
Ibk	84.1	72.1	82.1	69.7
J48	87.4	74.6	87.4	74.6
Multilayer Perceptron	87.4	68.9	87.4	70.5
Naïve Bayes	71.5	68.9	74.8	80.3
SMO	82.1	71.3	85.4	70.5

Table 3: Pre-University/First Term and Transfer/Non-transfer Classification Accuracies

Analysis

Examination of Table 2 shows that the accuracies tend to increase as students' academic careers develop. However, as mentioned earlier, the reliance upon later term data will not work in future testing of unknown data, as many students will be gone before they reach their third or fourth term. It is evident that the J48 (Decision Tree) and SMO (SVM) algorithms show the most stable and accurate predictions across these four semesters.

In Table 3, we find the use of only the data that is available as soon as possible in a student's academic career. It can be seen once again that the J48 algorithm has the highest accuracies across all four test sets, with the SMO algorithm close behind it. Also, a tremendous improvement can be seen in the Multilayer Perceptron (Neural Net) and Ibk (Lazy Learner) algorithms during *Term1*, which is most likely due to the split of transfer and non-transfer students. Another noticeable feature of Table 3 is that non-transfer classifications have a higher accuracy than transfer classifications. This is understandable, as many transfer students are likely to have a more diverse background than non-transfer students, which could cause unpredictable results. Table 3 also shows that the pre-university data has quite a significant impact on the overall accuracy of later term tests. One may conclude that the pre-university data is, in fact, more beneficial than the university data. In comparison to the studies mentioned in the related works section, these findings offer a good accuracy, a clearly-defined retention status, and a focus on early identification. Also, this approach consists of basic, easily-obtained data as well as a general data mining procedure, unlike some of the other studies.

Future Work

While this study relies only on a small amount of commonly available data, there are still other data sources that could be used. Current work is being done to add the data attributes in the *Student_Courses*, *Courses*, *Course_Terms*, and *Faculty* tables to the data warehouse. These elements are also very common at most higher-education institutions, as term grades must be kept for all students. Along with more data, it would also be useful to continue looking for better ways to increase classification accuracy, like through more precise preprocessing or better ways of dividing the test sets. Other algorithms could also be used in conjunction with the current algorithms, which may help increase accuracy. An algorithm combined with a decision tree, such as a hybrid model described by Carvalho (2004), could give greater classification accuracies, as the goal is to reduce as many errors as possible.

Another area of future work is the creation of well-defined rules that can be well understood by appropriate administration personnel. In addition, the generation of these rules will allow for the potential validation of these algorithms by others. Currently, we are working with TTU administrators that are interested in the work and its applicability to the entire university, and are interested in applying the algorithms to unknown students that are entering the university as freshmen.

Conclusion

It has been shown that through the use of various machine learning algorithms, decent classifications can be made for *Term0* and *Term1* students. In the future, the classification algorithms can be used with new students to predict which students will be retained until their senior year and which ones will not. Although the classification is not perfect, many of these students would be captured by the system described in this paper. The details of our data mining process - data cleansing, data warehouse attributes, data preprocessing, and data classification – are given in an attempt to make this process clear so that future work at other universities may benefit from our findings. We have stressed the generalization of data up until now, and this is necessary if institutions are to share similar data mining techniques. Without a similar database schema, there is no easy way to make comparisons between different studies.

Also, the simple data set keeps the inputs to the algorithms from being overly complex, which could lead to problems if the data is not processed correctly.

In the end, no amount of classification accuracy is going to keep students from leaving our program. It is student intervention that makes the difference. While intervention methods have not been mentioned in this paper, it is certain that those conducting intervention will need the most accurate results that they can get. It is believed that the results shown in this paper could outweigh current methods that counselors use to determine at-risk students. If that is true, these findings would help give those people the information needed to better target interventions, which could have a positive impact on student retention.

References

Aha, D. & Kibler, D. (1991). "Instance-based learning algorithms," *Machine Learning*. 6, 37-66.

Barker, K., Trafalis, T., & Rhoads, T. R. (2004). "Learning from Student Data," *Proceedings of the 2004 IEEE Systems and Information Engineering Design Symposium (SIEDS 2004).*

Carvalho, D. R., & Freitas, A. A. (2004, June 14). "A Hybrid Decision Tree/Genetic Algorithm Method for Data Mining," *Information Sciences*, 163(1-3), 13-35.

Chang, L. (2006, fall). "Applying Data Mining to Predict College Admissions Yield: A Case Study," *New Directions for Institutional Research*, 131, 53-68.

Dekker, G. W. (2009). "Predicting students drop out: a case study," *Educational Data Mining 2009: 2nd International Conference on Educational Data Mining, Proceedings (EDM 09)*.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten I.H. (2009). "The WEKA Data Mining Software: An Update," *SIGKDD Explorations*, 11, 1.

Han, J., & Kamber, M. (2006). *Data Mining: Concepts and Techniques*. San Francisco: Morgan Kaufmann Publishers, An Imprint of Elsevier.

Herzog, S. (2006, fall). "Estimating Student Retention and Degree-Completion Time: Decision Trees and Neural Networks Vis-à-Vis Regression," *New Directions for Institutional Research*, 131, 17-33.

John, G.H. & Langley, P. (1995). "Estimating Continuous Distributions in Bayesian Classifiers," *Eleventh Conference on Uncertainty in Artificial Intelligence*.

Luan, J. (2002, spring). "Data Mining and Its Applications in Higher Education," *New Directions for Institutional Research,* 2002(113), 17-36.

Mitchell, T. M. (1997). *Machine Learning.* New York: McGraw Hill.

Platt, J. (1998). "Fast Training of Support Vector Machines using Sequential Minimal Optimization," *Advances in Kernel Methods - Support Vector Learning*.

Quinlan, R. (1993). *C4.5: Programs for Machine Learning*. San Mateo: Morgan Kaufmann Publishers.

Sanjeev, A. P., & Zytkow, J. M. (1995). "Discovering Enrollment Knowledge in University Databases," *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD 95)*.

Yu, C. H., DiGangi, S., Jannasch-Pennell, A. & Kaprolet, C. (2010, April). "A Data Mining Approach for Identifying Predictors of Student Retention from Sophomore to Junior Year," *Journal of Data Science*, 8(2), 327-338.

Zhang, Y., Oussena, S., Clark, T., & Kim, H. (2010). "Use Data Mining to Improve Student Retention in Higher Education – A Case Study," *Proceedings of the 12th International Conference on Enterprise Information Systems (ICEIS 2010)*.