Using Decision Trees to Understand Student Data

Elizabeth Murray WUMPUS@OU.EDU
Basement of Carson, Room 36

Abstract

We apply and evaluate a decision tree algorithm to university records, producing human-readable graphs that are useful both for predicting graduation, and understanding factors that lead to graduation. We compare this method to that of nueral networks, Support Vector Machines, and Kernel Regression, and show that it is equally powerful as a classificaion tool.

At the same time, decision trees provide simple, readable models of graduation that we hope decision-makers will find useful in assessing their programs and understanding their student body.

1. Introduction

Universities generally possess large bodies of both attitudinal and demographic student data. This data is a wealth of information, but is too large for any one person to understand in its entierety. Understanding salient characteristics of these data and how they fit into current models of retention and graduation is an essential task in education research, and is part of a larger task of developing programs that increase retention, graduation, and student learning.

Generally (at least, at this university), this type of data is presented to decision makers in the form of tables or charts, and without any substantive analysis. Most analysis of the data is done according to individual intuition, or is interpreted based on prior research.

A typical analysis might involve expert examination of large tables of statistics, such as graduation or retention percentages. The analysis depends largely on the expertise of the individual performing analysis, the question the expert is seeking to answer, and the expert's past experience.

Appearing in *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, 2005.

When formal analysis of the data *is* performed, it is generally to find a way to predict graduation. Logistic regression is a common method of analysis (Chao-Ying) for these types of data sets, although other methods have been studied with some success.

For this data set, no algorithm has been able to correctly classify students. It is possible that the current surveys and records do not provide enough information for good classification. Nonetheless, previous studies of this data set (Barker, 2004) have revealed interesting aspects of the data, such as the effects of math readiness and hometown population on graduation probability.

This paper deals with two related problems: using this type of data to predict whether or not a college student will graduate within six years, and transforming the data into meaningful visual structures that decision makers can use to guide their intuition. The latter problem is the main focus, while the former is examined to compare the effectiveness of the decision tree algorithm to other data mining techniques.

2. Problem Statement

The University of Oklahoma collects data about their students in two ways: via the mainframe database that stores grades and transcripts, and through an attitudinal [1] survey of all incoming freshman. Analyzing these collections of data (both very large, and sometimes incomplete or damaged) can help educators and adminstrators identify high-risk students who are not likely to graduate, and exceptional students who are very likely to graduate. This, in turn, can help them decide where to spend resources – either to help high-risk students or entice exceptional students.

For this semester project, I chose to analyze both data

[1] The attitudinal survey asks students for their opinions about the university and about themsleves. For example, the survey asks a student whether they believe they will succeed in college. It also asks them whether their parents went to college – but this information is not externally verified.

sets, individually and together, to see if I could predict graduation and determine the factors that most influenced this prediction. I chose to use decision trees for this project because they are simple (and therefore maintainable by whoever will take my job when I leave it – and he or she will probably not have taken more than one basic programming course) and because they are easy to explain in plain english, or in graphical formats that adminstrators can understand without an understanding of the algorithm itself.

It is not surprising that this problem has been extensively studied, since universities devote significant resources to seducing likely graduates to their programs, and helping high-risk students. In particular, the University of Oklahoma gives National Merit scholars a full scholarship and minority engineers also recieve a scholarship and access to additional support networks.

Previous research (Barker, 2004) suggests that classifying incoming students as either graduates or non-graduates, given the current data, is difficult. Intuitively, one would expect that earning a degree involves not only intelligence and academic preparedness, but perserverence, luck, social involvement, and university atmosphere. In fact, popular models of dropout take such factors into account. Such attributes are difficult to discern, and even more difficult to quantify, and have been the subject of higher education research for several years.

What can be learned from a data set that does not provide enough information for accurate predictions? At least, we can see how student characteristics influence graduation. Decision trees give probabilities of graduation, and use probability thresholds (i.e. classifying all students with a probability of graduation above 0.50 as graduates) to make classification decisions. These probabilities, while never exactly 1 or 0, still contain valuable information about how certain attribues influence graduation.

The standard method for evaluating probability trees is to use them as classifiers, and measure their True Positive Rate (TPR) and False Negative Rate (FNR) on all possible thresholds. These rates are then used to draw a Reciever Operator Curve (ROC) and compute the area under this curve (AUC). This method of validation is employed in this paper, as are additional methods that attempt to determine the accuracies of the probabilities given by tree.

3. The Data

As mentioned above, there are two sources of data: the university mainframe database, and a survey of all incoming freshman.

3.1. University Mainframe Database

The University Mainframe Database contains four major tables:

- A *Student Table* containing one entry for every student ever enrolled at the University of Oklahoma. This table contains SAT Scores, ACT Scores, and other pre-college information.
- A *Semester Table* containing one entry for every semester than every student was ever enrolled. From this table, one can determine a student's overall GPA, and GPA for every semester, as well as the number of credits the student earned and when they earned them.
- A *Course Table* containing one entry for every time a student has ever taken a course at the unviersity. From this table, one can determine a student's initial math level, and grades in individual courses.
- A *Scholarship Table* that contains one entry for every scholarship that any student has recieved.

3.2. Survey of Incoming Freshman

Incoming freshman are surveyed informally by University College. We stress that data collection is informal, and intended for use internal to the university. Regardless, this data set provides good information that aids in classification.

Survey results (which do not exist for all students, and are not complete for all students for which they exist) are available for the years 1995, 1996, and 1997. While the surveys for each year are different, there are fifty-two questions common to all years.

Here are some sample questions from the survey:

```
"In high school, I met as many people
and made as many friends as I would
have liked."

"It is a) Extremely Important,
b) Important, c) Relatively Unimportant,
or d) Totally unimportant to gain a
background for lifelong learning while
I'm at OU."
```

4. Previous Work on this Problem

Several data mining techniques have been applied to the problem of modeling graduation. A review of at-

tempts at using logistic regression to model graduation can be found in (Chao-Ying). Other studies have used Survival Analysis to develop Proportional Hazards Regression Models. The use of decision trees, in particular, has been studied at Oregon State University. Little research has been done on the usability of particular methods, or the integration of predictions into software designed to aid administrators in understanding student retention.

Previous attempts to model *this data set* are of particular relevance. Kash Barker wrote his master's thesis on his attempts to predict graduation from the University College student survey. He used neural networks and support vector machines to predict student graduation. He achieved anywhere from 36% to 40% misclassification rates, which is an improvement over random given that the default six-year university graduation rate is approximately 50% (for the students that took the survey in 1995, 1996, and 1997).

The FPR and FNR values are not available for his experiments.

5. The Decision Tree Algorithm

Decision Trees are an intuitive and widely-used type of influence diagram. The basic goal of a decision tree is to find an optimal set of yes-or-no questions that ultimately leads to a correct classification, or probability. The tree must have meaningful criteria for choosing questions, and derives answers from the training data set.

Tree construction is recursive. We begin with some large data set and, through some predefined method, select some question about each data item that will 'split' the data. For our data set, a potential question might be "Does the student have an SAT Score above 1300?" This question divides the data set into two parts: those students with an SAT Score above 1300, and those without. We then continue the process for these two groups of students, and for any subgroups we generate from them.

How should one choose the question? Common methods of choosing the question are:

- Choose the question so that the two groups are significantly different.
- Choose the question to minimize the entropy of the two groups.
- Some combination of the above two methods.

In this experiment, questions were chosen to minimize entropy, but the division was also required to be 99.99% significant, according to the Chi-Squared distribution. No group of students was ever smaller than twenty, as the Chi-Squared statistic is not accurate for fewer than twenty samples.

Additionally, groups of students were never split in such a way that the sampling error greater than 5%, with 95% confidence. This additional criteria was added to ensure that the probabilities at the node were accurate, as one of our main goals was to ensure that the model was comprehensible to humans.

The final algorithm is as follows:

```
INPUT: a list of binary strings. The
first bit of the string corresponds
to a "TRUE (GRADUATE)" or "FALSE
(NON-GRADUATE)" classification. The
remaining bits are attributes.

OUTPUT: A decision tree.

STEP 1: Create the first node of the
tree. This node contains all students,
and the probability of graduation is
equal to the overall graduation rate
for the set of students:
    GRADUATES/(GRADUATES + NON-GRADUATES).

STEP 2: Push this node onto an empty
        stack, S
STEP 3: Create an empty list L of
        completed rules

WHILE (S is not empty)

   IF (there is some way to split the
   data set on top of the stack into
   parts A and B s.t. A and B are
   different with 99.99\% confidence,
   as tested with the chi-squared
   statistic AND the sampling error is
   smaller than 0.05 AND there are at
   least twenty samples corresponding
   to this rule in the training set)

      Find the most significant way to
      split the data, creating parts
      A' and B'. Push A' and B' onto
      the stack S

   ELSE
      Add the data on top of the
```

```
      stack to L


OUTPUT  the set of nodes/rules L, and
        the splits that created them.
        These splits correspond to
        decision rules.

// Since each new rule is a
// refinement of another rule, the
// rules form a tree.
```

This was implemented as a Java program that accessed a MySQL database. One table was used for training, and another for testing. Each rule corresponded to a MySQL query that drew a set of students from the training table. To test the tree, the query was modified to draw students from the testing table. When the query was run on each table, the proportion of students from either table could be compared.

The general algorithm described above can be modified in a few ways:

- The significance required for a split can be lowered or raised.
- The sample-size at each node can be tweaked – generally to make sure that the sampling error at a node is within some acceptable range.
- Rather than computing gain using both sides of a split, we could use just one side, introducing a bias for more dramatic trees. While such trees might be worse for classifying graduates, they could concievably be better for data exploration.

I chose the confidence level of 99.99% and the sampling error of 0.05 because they performed well on the test sets. Smaller statistical significance, even 90% or 95%, tended to overfit the data; a tree that used smaller significance to determine splits generally performed better on the training set than on the test set, although it is worth noting that they all performed equally well on the test set.

Other parameters could give more readable trees (by generating shorter sets of questions) and equally accurate probabilites, but may give poorer misclassification rates. However, when one considers that poor misclassification rates generally result from a large number of nodes with probabilities close to $\frac{1}{2}$, one realizes that they are not necessarily useless: if a tree contains even one group of students that can be reliably said to graduate at rates significantly higher or lower than average, then the tree has discovered something interesting.

5.1. Processing the Data for input

Most survey questions are based on some kind of Likert scale[2]. If we considered an attribute to be "an answer to a survey question" then we would have attributes that could take on more than two values. Instead, each survey question corresponds to several values. Before the question is inputted into the algorithm, it is converted into several binary-valued attributes.

While decision trees are fully capable of handling multi-valued questions, it is conceptually simpler to avoid them, and *also does not permit the tree to split on the same question twice, even if it chooses a different value for that question.* We could just have easily used gain and the chi-squared statistic with multiple-valued questions, but chose not to because we wanted to give the tree as large a search space as possible.

For example, the first question listed above would be ranked by the student on a scale from one to ten. This question corresponds to ten binary valued attributes:

- "The student ranked the importance of the question as 1 or greater." T/F
- "The student ranked the importance of the question as 2 or greater." T/F

  .

  .

  .
- "The student ranked the importance of the question as 9 or greater." T/F

This increases the number of attributes, but only by a factor of at most ten, and allows the algorithm to select attributes that correspond to intervals. For example, to seperate instances along some interval the algorithm can seperate the data on a "greater than 2" attribute, and a "greater than 5" attribute. This seperates the data into three groups: those less than 2, those between 2 and 5, and those greater than 5.

Likewise, SAT scores and highschool GPAs are processed into interval attributes, such as "SAT Score

[2] A Likert scale is a standard question type in surveys, and the reader is undoubtably familiar with it. A Likert scale asks the questionee to rank their answer to a question on some scale that ranges from one extreme opinion to another. For example, a Likert Scale might ask you to rank your confidence in the current president from "Completely Confident" to "I'm not sure" to "No Confidence whatsoever."

> 1000" or "SAT Math Score > 600." Again, this allows the algorithm to select optimal intervals (width $= 100$), rather than forcing it to split the tree on every possible SAT Score value.

5.2. Removing Incomplete Data

Barker removed from his data set all incomplete surveys or surveys completed by students with identification numbers that could not be found in the mainframe database. For the sake of comparison, the decision tree described here was tested on a data set that had undergone similar preparation. [3] If a student left any question blank, they were removed entirely from the data set before the data set was inputted into the algorithm. This reduced the number of students from 7000 to 5075.[4]

6. Validation Methods

Barker tested his chosen algorithms in two different ways (the terms are his):

- Between Years Testing: Use one year of data to train, and a different year of data to test. The training set year is always smaller than the testing set year. His results are shown in Table 1.
- Among Years Testing: Use 70% of the data from a particular year to train, and the remaining 30% to test. The results of Barker's Among Years tests are shown in Table 2

For comparison, I performed Between Years testing using the decision tree algorithm. I also tried training the tree using two years, and testing on the remaining year, and used this to compare to Barker's Among Years tests.

6.1. Results of Validation

The ROC and it's corresponding AUC are well-known measures of probability tree learning. On average, over five between years tests, the decision tree misclassified 39.3% of the testing set, and 38.6% of the testing set in the among years tests. The average AUC for the between years test was 0.64, and 0.65 for the among years test. The average misclassification rate for the algorithms that Barker tested was 38.3%. The difference in misclassification rate is small, and we therefore conclude that decision trees perform as well as neural networks, support vector machines, and kernel regression, when they are used as a classifier.

Additionally, we note that the variance in our misclassification rates is *much* smaller, and that the trees presented here performed equally well on their testing and training sets. See (Barker, 2004) for a comparison of Barker's training set misclassification rates and testing set misclassification rates.

The detailed results of the Between Year tests are shown in Table 3. The results of tests that used two years of data for training, and one year for testing are shown in Table 4.

All of these tables show the "probability threshold" for classifying a student as a graduate, the overall misclassification rate, the false positive rate, and the false negative rate for each threshold and training/testing set pair. The probability threshold is the proportion of positive training examples that must correspond to a particular rule (or leaf node) in order for all samples at that node to be classified as "positive."

To further test the accuracy of a probability tree (as opposed to a tree used for classification), I trained a tree, and then distributed a test set through the tree. Once the test set has been distributed through the tree, each node will hold four values: the number of positives and negatives from the test set and the number of positives and negatives in the training set. We want to answer the following question for a particular node: is the difference between the ratio of training positives to training negatives significantly different from the ratio of testing postives to testing negatives?

This is a non-standard method of validation, and apparently requires a detailed explanation. Suppose that you are given two probability trees, A and B. At each leaf node, A contains some probability, and A has an AUC of 0.60. B is the same as A, except the leaf nodes of B have an associated probablity of 1 when the corresponding leaf of A has a probability of 0.50 or larger, and 0 otherwise.

Notice that A and B are equally good for the purposes of classification (since a threshold of 0.5 will always given the best classification rate), and will have an AUC that is virtually the same. But it is not accurate to say that they contain the same amount of knowledge. A clearly knows more than B, since A can give more accurate classifications for specific groups of students. A can also estimate rates, but B can only

[3] It is worth noting that a premilinary test on the complete data set yielded much better accuracy, which seems to indicate that completion of the survey is an attribute in itself. This suggests that blank survey questions are not the result of an input error, despite the fact that survey responses are entered into a computer by hand.

[4] Barker, after cleaning his data, had 5100 students. I cannot explain the difference, but twenty five students is probably not enough to invalidate the comparison.

give classifications.

The AUC is essentially a measure of learning *for the sake of classification.* We are interested in finding not only the accuracy of the classifaction, but the accuracy of the claimed influence of the decision rules on graduation rates. We want to know whether the data from the testing set bears some structural resemblence to the data from the training set.

For each node, I ran a Chi-Square test to reject or accept this null hypothesis. I then counted the fraction of times that the test failed (failure is good, in this case, given that accepting H_O means there is a significant difference between training and testing probabilites) and used this to measure the accuracy of the probability.

The results of this test are shown in shown in each table. In general, one or two nodes were rejected.

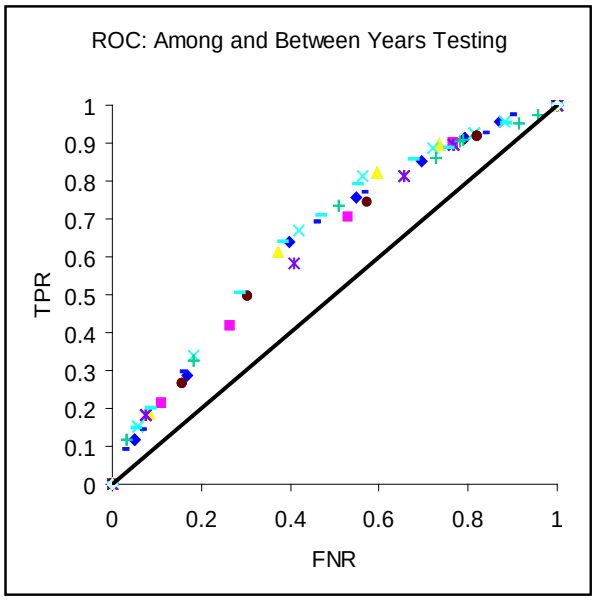


Figure 1. ROC for Among & Between Years testing of the Decision Tree. The black line is y=x, and is the ROC for the random algorithm.

7. Estimating a Graduation Rate

The test trees (both Between Years and Among Years) were used to estimate graduation rates for their corresponding test data sets. The predicted graduation rate and actual graduation rate are shown in Table 5. They are unremarkable, and we include them just to satisfy the reader's curiosity, since the corresponding question is obvious.

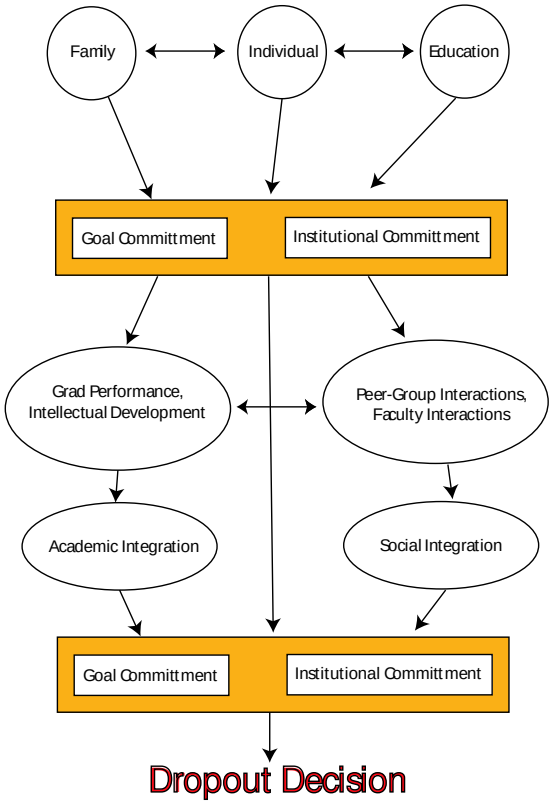


Figure 2. Tinto's model for graduation.

8. Graphical Representation of the Tree

A good graphical representation of the tree is practically important. Figure 3 shows the decision tree constructed from the 1995 and 1996 data, and was tested on the 1997 data. The number of nodes is small enough to be legible.

An interesting method of display (not really related to machine learning, interesting nonetheless) for a decision tree is the Decision Ring representation. The Decision Ring uses the probability wheel concept to represent probabilities in an intuitive way. This is essentially a pie-chart representation of the probabilities, and is described in (Bordley).

This representation has the added advantage of avoiding overly precise numbers – the tree cannot really claim that 'precisely 77.2%' of a certain group of students will graduate. The decision ring representation prevents the user from taking the numbers too literally.

9. Discussion & Future Work

Frank and Witten (Witten) present a permutation test for determining a split, arguing that the chi-squared test is not only inaccurate for small sample sizes, but may not even be an accurate distribution to begin with. They propose a Monte Carlo-style algorithm for approximating the true distribution. It is possible that using a permutation test could increase the accuracy of the algorithm.

However, even an improvement of 5%-10% in the misclassification rate would not make the the algorithm much more useful for classification. A more hopeful question: can we get a better *model* of graduation.

Decision trees are not a truly satisfying model, for several reasons:

1. The model does not express the interaction of attributes very well;
2. The model is not expressive enough, as all rules are simply yes-or-no questions. In the tree, attributes are either true or false. It is true that some attributes are the encoding of spectrum of possible answers, but even then the tree is forced to pick some cutoff interval; and
3. The model does not tell us where we are missing data. Obviously, we assume that it is possible to predict graduation, given the right data – but do we have it? The tree does not say either way – nor does it give us clues as to where we might improve data collection.

In the future, using Bayes nets to model graduation might prove more fruitful. In fact, the most widely accepted model for understanding graduation is Vincent Tinto's model, based on psychological theories of suicide.

The reader will also notice that Tinto's model looks very much like a Bayes net, and not just superficially: the model shows student attributes, the interplay between them, and their influence on graduation. Moreover, building a Bayes Net based on Tinto's model would take advantage of three decades of research in higher education, and is therefore a very reliable source of expert knowledge.

Specifically, the Bayes net would resemble Tinto's model exactly, with additional nodes for each attitudinal survey question, and for the pre-college academic variables. These additional nodes would be connected to either the Family, Individual, or Academic nodes, and the search space would be the set of all possible Conditional Probability Tables for the network. This search space can be searched in several ways, notably by EM and Gradient algorithms.

By computing the likelihood the net (given the data), a Bayesian network can be used to test the likelihood of a theory. This would provide administrators with the ability to test their own theories, perhaps allowing them to discover models for retention and student satisfaction, in addition to graduation. Exploring good graphical representations of Bayesian networks would also be very useful.

It is hard to say whether a Bayes Net based on Tinto's model (or some other source of expert knowledge) would produce lower misclassification rates, but it may pick up where decision trees fell short: it might help us to better understand the data we have collected, point out shortcomings in the current data, and help us to understand how we can help students reach graduation.

10. Conclusion

The misclassification rates given by these types of decision trees are not better than those Barker achieved using Nueral Networks, Support Vector Machines, Kernel Regression, and, most recently, Logistic Regression. The main benefit in using this method is that we can achieve the same accuracy using only a handful of rules. In fact, the test trees used about eight rules on the average.

Aside from being human readable, these trees give fairly accurate probabilities of graduation. Most of the time, the graduation rates given for a leaf in the tree are not significantly different from the corresponding group of students in the test set. If it is not possible to perfectly classify students based on this data, then at least we want to know which attributes increase or decrease the probability of graduation, and how much affect they have. Given that expert intuition is of the utmost importance in higher education research, the degree to which experts can read the tree is also a very important factor in selecting a data mining algorithm for student data.

The AUCs for the tree are low, but they are invariably better than random. The tree has learned something, and we can get access to what it has learned via the probabilites.

Decision trees are not better for classification than previously tested algorithms, but they are simple to implement, human-readable, and can give partial information about how certain pre-college attributes affect

graduation. In these respects, they are superior to other methods.

Given that they are enormously simple to generate (i.e. they are free, since code now exists to generate them from the database), I recommend that decision trees become a new tool for student data analysis in the College of Engineering. Informal tests of the tree on engineering cohorts shows that even without attitudinal data, they are 36% accurate (with an AUC of 0.65) in predicting graduation in engineering students. They should always be displayed carefully (perhaps using decision rings or probability wheels) to prevent overly literal interpretations, such as interpreting the trees as definitive models of graduation.

References

Barker, K. (2004). *Learning From Student Data* Masters Thesis, Department of Industrial Engineering, University of Oklahoma.

Mitchell, Tom. (2004). *Machine Learning* McGraw Hill, 1997

Bordley, Robert F. *Decision Rings: Making Decision Trees Visual and Non-Mathematical* INFORMS Transactions on Education 2:3

Joanne Chao-Ying, Harry Tak-Shing, Frances Stage, Edward St. John *The Use and Interpretation of Logistic Regression in Higher Education Journals*

Frank & Witten *Using a Permutation Test for attribute selection in Decision Trees*

Table 1. Between years misclassification rates from Barker's thesis.

Algorithm	Train/Test	Mis. Rate
Fischer's Discriminant	1995/1996	38.6%
Fischer's Discriminant	1996/1997	37.3%
Perceptron Algorithm	1995/1996	40.3%
Perceptron Algorithm	1996/1997	42.4%
Neural Net	1995/1996	39.6%
Neural Net	1996/1997	38.4%
Support Vector - Linear	1996/1997	38.7%
Support Vector - Linear	1996/1997	37.9%
Support Vector - Polynomial	1996/1997	38.8%
Support Vector - Polynomial	1996/1997	38.0%
Support Vector - Radial Basis	1996/1997	37.7%
Support Vector - Radial Basis	1996/1997	37.9%

Table 2. Among Years misclassification rates from Barker's thesis.

Algorithm	Mis. Rate
Fischer's Discriminant	39.1%
Fischer's Discriminant	40.4%
Fischer's Discriminant	35.5%
Average	38.3%
Perceptron Algorithm	39.3%
Perceptron Algorithm	39.2%
Perceptron Algorithm	40.2%
Average	39.6%
Neural Net	39.2%
Neural Net	40.2%
Neural Net	36.9%
Average	38.8%
Support Vector - Linear	38.4%
Support Vector - Linear	40.1%
Support Vector - Linear	35.9%
Average	38.1%
Support Vector - Polynomial	38.4%
Support Vector - Polynomial	40.1%
Support Vector - Polynomial	36.1%
Average	38.2%
Support Vector - Radial Basis	37.9%
Support Vector - Radial Basis	38.2%
Support Vector - Radial Basis	34.6%
Average	36.9%

Table 3. Misclassification rates for Between Years testing of the Chi-Squared Decision Tree (training set pop. approx. = 1600, testing set pop. approx. = 1600).

Train/Test	Thresh.	Mis. Rate	FPR	FNR
1995/1996	0.25	47%	9%	79%
1995/1996	0.50	38%	36%	40%
1995/1996	0.75	45%	100%	0%
AUC	0.642			
Rejected	12.5%			
1996/1997	0.25	47%	10%	77%
1996/1997	0.50	41%	58%	27%
1996/1997	0.75	45%	100%	0%
AUC	0.619			
Rejected	20.0%			
1997/1996	0.25	42%	11%	73%
1997/1996	0.50	38%	38%	37%
1997/1996	0.75	49%	100%	0%
AUC	0.655			
Rejected	20.0%			
1995/1997	0.25	45%	8%	81%
1995/1997	0.50	38%	33%	42%
1995/1997	0.75	49%	100%	0%
AUC	0.659			
Rejected	37.5%			
1996/1995	0.25	45%	11%	77%
1996/1995	0.50	41%	42%	41%
1996/1995	0.75	47%	100%	0%
AUC	0.624			
Rejected	0.0%			
1997/1995	0.25	47%	8%	82%
1997/1995	0.50	40%	51%	30%
1997/1995	0.75	47%	100%	0%
AUC	0.620			
Rejected	40.0%			
Average	0.25	45.6%	9.3%	78.2%
Average	0.50	39.3%	43.1%	36.1%
Average	0.75	47.3%	100%	0%
AUC	0.637			
Rejected	21.7%			

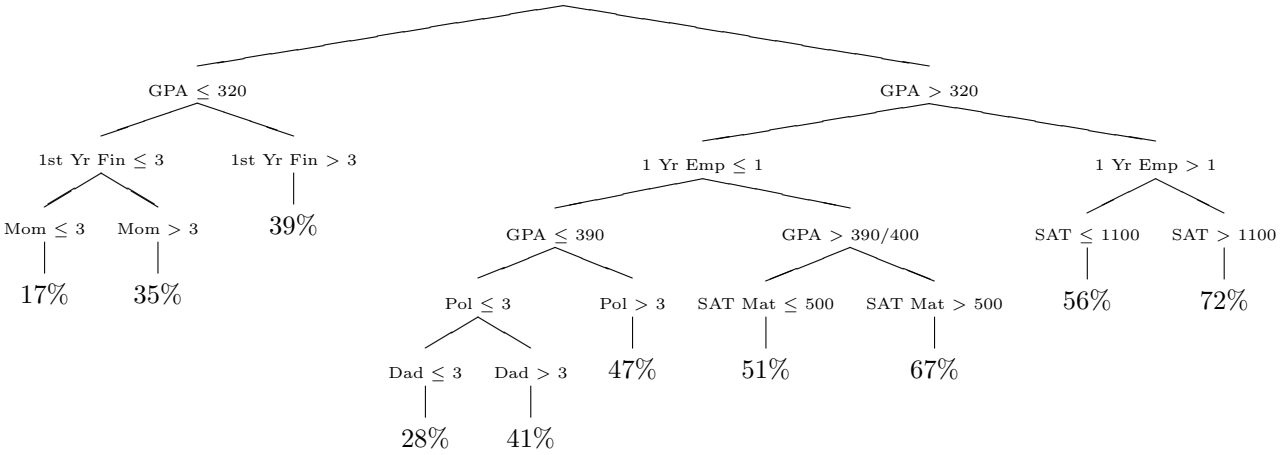
Table 4. Misclassification rates for Among Years testing of the Chi-Squared Decision Tree (training set pop. approx = 3200, testing set pop. approx. = 1600).

Train/Test	Thresh.	Mis. Rate	FPR	FNR
1996-7/1995	0.25	46%	9%	79%
1996-7/1995	0.50	39%	27%	51%
1996-7/1995	0.75	43%	88%	4%
AUC	0.637			
Rejected	30.8%			
1995-7/1996	0.25	50%	3%	89%
1995-7/1996	0.50	39%	31%	46%
1995-7/1996	0.75	43%	91%	3%
AUC	0.644			
Rejected	16.7%			
1995-6/1997	0.25	47%	5%	88%
1995-6/1997	0.50	37%	36%	38%
1995-6/1997	0.75	49%	100%	0%
AUC	0.658			
Rejected	10.0%			
Average	0.25	47.8%	5.4%	85.8%
Average	0.50	38.6%	31.2%	45.0%
Average	0.75	45.1%	92.9%	1.9%
AUC	0.646			
Rejected	19.2%			

Table 5. Graduation Rate Predictions from the Decision Tree.

Train/Test	Predicted	Actual
1995/1996	46.2%	45.3%
1997/1996	47.8%	45.3%
1996/1997	46.6%	49.2%
1995/1997	47.7%	49.2%
1996/1995	46.9%	47.3%
1997/1995	50.0%	47.3%
1996-7/1995	47.5%	47.3%
1995-7/1996	45.9%	45.3%
1995-6/1997	46.8%	49.2%

Figure 3. The Decision Tree generated from the 1995 and 1996 data sets. The leaves of the tree represent graduation rates for students who fell into those leaves in the training set. When tested on the 1997 data set, only 10.0% (1) of the leaves were found to have statistically significantly different graduation rates in the training and test sets. When classifying students who fell in a leaf with more than 50% graduates as "graduates," the tree was 63% accurate, 64% accurate classifying graduates, and 62% accurate classifying non-graduates.



Here is is a list of the questions abbreviated in the tree:

- GPA: High school GPA, from 0 to 400.
- 1st Yr Fin: "At present time, I have enough financial resources to complete my first year at OU. 1) Strongly Agree 2) Agree 3) Nuetral 4) Disagree 5) Strongly disagree."
- Mom: "My Mother: 1) Did not complete high school 2) Graduated from highschool 3) Did some college work 4) Recieved a bachelor's degree 5) Recieved a degree beyond a bachelor's degree."
- 1st Yr Emp: "I need to work to afford to go to school. 1) Strongly Agree 2) Agree 3) Nuetral 4) Disagree 5) Strongly disagree"
- Pol: "I would characterize my political beliefs as: 1) Very Liberal 2) Liberal 3) Middle-of-the-road 4) Conservative 5) Very Conservative."
- Dad: "My Father: 1) Did not complete high school 2) Graduated from highschool 3) Did some college work 4) Recieved a bachelor's degree 5) Recieved a degree beyond a bachelor's degree."
- SAT: Scholastic Achievement Test score (Students who did not take the SAT have a score of zero).
- SAT Mat: Score on the SAT Math section (Students who did not take the SAT have a score of zero).