Building Individualized Course Outcome Models to Enhance Student Success: A Primer

State University of New York University Faculty Senate Undergraduate Programs and Policies Committee

by Ronald Sarner
SUNY Polytechnic Institute
March, 2020

Introduction:

As academic advisors we frequently have to make judgments regarding how likely particular students are to succeed in a course. We have some tools available to us, notably prior academic records and test scores, and we make recommendations based upon those. We also have anecdotal accounts from previous students to the extent that we remember them and correctly recall them. To some extent wisdom and knowledge accrues with age, and, thus, junior faculty serving as academic advisors may be at a decided disadvantage *vis-à-vis* senior faculty. Moreover, over time student preparation changes and likewise courses are altered. Advice that might have been good at one point in time may not be so accurate at another time. As colleges become more attuned to reducing attrition, quality advisement becomes increasingly important.

In my own discipline, computer science, I have often heard well-intentioned advisors suggest that success in introductory programming courses requires a good background in mathematics. As a person who has taught introductory courses for decades, I am somewhat taken aback because I know that most instructors do not use examples requiring anything more complicated than ninth-grade algebra. So, if performance in math is not a good predictor of success in an introductory programming, what is? I know that in my institution, over the past decade or so, about 30% of students who enroll in an introductory programming course do not perform well; they get a grade lower than a full "C" or they withdraw. Despite my decades of teaching the course, on the basis of my observations I am incapable of discerning the measurable characteristics that separate students who succeed from those who do not. While I may be incapable of discernment on the basis of observation, data analytics is not so hampered. With historical data in hand, it is entirely possible to use its tools to describe the characteristics that separate those who succeed from those who do not. Moreover, the tools are not only able to determine that a particular indicator, for example high school average, is a good predictor of risk in a specific course, but will also identify the cut point that separates high-risk students from low-risk students – a degree of precision that no advisor relying on anecdotal examples can match.

Descriptive models can be created for many courses; at SUNY Poly we build them for courses typically taken by incoming freshmen. These models can then be used to evaluate the risk of an adverse outcome for any given student in every course typically taken by first-year students. By no means should students be discouraged from enrolling in challenging courses – but in this era when all colleges are concerned about attrition rates it behooves us, where possible, not to advise them into multiple high-risk courses in the same semester, particularly in that crucial first semester of enrollment. In many of today's majors academic plans are so structured that students are left with few choices, but even in these situations students are left with choices regarding how to satisfy particular general education requirements.

Using readily available, contemporary data-analysis tools, it is both possible and desirable for colleges and universities to build and use models that provide personalized predictions of student risk and success in particular courses.

While these costly services are touted and marketed by several firms, the process is one that should be within the skill set of institutional research staff on all of our campuses.

The purpose of this primer is to provide a step-by-step guide that makes it possible for campus personnel to build individualized student predictions based upon their own campus performance history

and to do so in a few person-hours of effort and at little (and probably no) cost. The process described here is the one in use at SUNY-Poly which has a small undergraduate population with a limited number of majors. Readers should keep in mind that the process will have to be adapted to institutions that are larger, or with a more comprehensive suite of programs. Additionally, while the illustrations are correct for the versions of the software that were used (Excel and SPSS), subsequent changes to either may alter the menu choices shown.

Requirements:

- ability to extract needed student performance and demographic data from the student records system
- access to Excel
- access to SPSS (Statistical Package for the Social Sciences) including Decision Trees

To verify that your version of SPSS includes Decision Trees, go to the **Analyze** menu; the **Classify** option should not be grayed out. Select it. On the sub-menu the **Trees** option should not be grayed out. If **Trees** is present and not greyed out you have an appropriate version.

Initial Data Preparation

Select a time period for this study. At SUNY-Poly we typically use the last five academic years (excluding summer or other extraordinary terms). With our typical freshman class of about 300, a five-year period is reasonable in order to get a sufficient number of cases; for a large campus extracting five years of data would be overkill. In any event, it is necessary to produce two files – one containing demographic data – one record (one line of data) for every student enrolled during that time frame and the second file containing one record (one line of data) for every individual course enrollment. Institutional Research staff at SUNY-Poly will make scripts available to extract this data to other SUNY campuses using Banner upon request.

Assuming the task at hand is to build predictive models for incoming freshmen, the following data elements should be extracted to form the demographic file:

- ID number – whatever the campus uses for student identification
- Term matriculated (or admitted) – will be used to determine whether a course was taken in the first semester of the first college year
- high school average
- SAT verbal score (or converted ACT equivalent)
- SAT quantitative score (or converted ACT equivalent)
- major (if first-year students are permitted to declare a major)
- on-campus housing (yes/no)

Additional data elements can be included but experience at SUNY-Poly is that these generally do not enter into the predictive models. Again, local adaptations may be necessary. Community colleges, for example, do not typically have SAT or ACT scores. Depending upon the timeframe being used there may be a concern that the nature of the measure (for example changes in the SAT) has changed and that an identical score in two different years may not be comparable.

If there is more than one record per student, duplicate records must be removed; this can be done in Excel (not illustrated here). Again, adaptations may have to be made if the number of records is so large as to make the use of Excel impractical or impossible. The demographic data is stored in an Excel spreadsheet as illustrated in Figure 1. Student ID numbers have been anonymized to preserve confidentiality.

	PSEUDOID	Term_Code_Matric	Housing_Ind	Major_Desc	SAT_Verbal	SAT_Math	High_School_GPA
2630	2630	201409	Y	Network and Computer Security	640	750	84.0000
2631	2631	201409	N	Business Administration	370	400	86.2000
2632	2632	201409	N	Business Administration	.	.	.
2633	2633	201401	N	Business Administration	570	570	.
2634	2634	201401	N	Health Information Management	.	.	.
2635	2635	201409	Y	Computer & Information Science	580	620	95.7500
2636	2636	201401	N	Computer & Information Science	.	.	.
2637	2637	201401	N	Technology Management	.	.	.
2638	2638	201401	N	Community & Behavioral Health	.	.	.
2639	2639	201409	N	Computer & Information Science	.	.	.
2640	2640	201509	N	Psychology	.	.	.
2641	2641	201401	N	Computer & Information Science	.	.	.
2642	2642	201509	Y	Computer & Information Science	470	540	86.1000
2643	2643	201409	N	Undeclared Major	560	520	92.2800
2644	2644	201409	Y	Computer & Information Science	580	660	86.7000
2645	2645	201401	N	Computer Information System	.	.	.
2646	2646	201409	Y	Computer/Info Sci-BS/MS	690	750	84.5000
2647	2647	.	N	Major Not Declared	450	480	86.5900
2648	2648	201409	Y	Mechanical Engineering	450	560	87.7600
2649	2649	.	N	Major Not Declared	480	420	92.9000
2650	2650	201401	N	Communication/Informatn Design	.	.	.
2651	2651	.	N	Major Not Declared	390	510	.
2652	2652	201401	N	Business Administration	.	.	.
2653	2653	201409	N	Business Administration	.	.	.
2654	2654	201409	N	Health Information Management	.	.	.
2655	2655	201409	Y	Business Administration	420	410	89.6000
2656	2656	201409	Y	Mechanical Engineering Tech	500	660	88.9200
2657	2657	201409	Y	Undeclared Major	460	520	85.0000
2658	2658	201401	N	Health Information Management	.	.	81.3200

Figure 1: Demographic Data File (duplicate ID numbers have been removed)

The second file consists of one record for each course enrollment. Thus, a student taking five courses in one semester would generate five records for that semester. At a minimum each record must contain:

- ID Number in the same format used for the Demographic file
- Semester ID – the semester the course was taken, for example 201709 is Fall 2017
- Course Number
- Grade

Additional data elements could be added. For example if class start time is added (e.g. 0800, or 1200, or 1800) it would be possible to determine if students with particular characteristics perform better in classes starting at a certain time of day – or to put it more crudely, are there characteristics of "morning people", or "afternoon people"? Adding the name of the instructor would make it possible to determine if an instructor has better outcomes with students of a certain preparation, though this level of analysis begins to raise ethical issues.

A sample of part of the grade file is shown in Figure 2.

	Term	PSEUDOID	Course	Final_Grade
30069	201501	2629	CS-543	F
30070	201506	2629	CS-543	F
30071	201409	2630	CS-100	A-
30072	201409	2630	FYS-101	S
30073	201409	2630	MAT-112	F
30074	201409	2630	NCS-181	B-
30075	201409	2630	PHI-130	D+
30076	201501	2630	CS-108	W
30077	201501	2630	HIS-101	B
30078	201501	2630	IDS-103	C
30079	201501	2630	STA-100	F
30080	201509	2630	COM-106	B+
30081	201509	2630	MAT-112	A
30082	201509	2630	STA-100	C+
30083	201601	2630	CHI-101	A
30084	201601	2630	CS-108	B-
30085	201601	2630	MAT-115	B-
30086	201609	2630	NCS-210	W
30087	201609	2630	PHY-101L	W
30088	201609	2630	PHY-101T	W
30089	201609	2630	SOC-100	W
30090	201701	2630	COM-306	B+
30091	201701	2630	NCS-205	A
30092	201701	2630	PSY-100	B
30093	201709	2630	IS-310	D
30094	201709	2630	NCS-210	F
30095	201709	2630	PHY-101L	C+
30096	201709	2630	PHY-101T	F
30097	201801	2630	NCS-315	F
30098	201801	2630	NCS-320	A-
30099	201801	2630	NCS-330	D
30100	201409	2631	BUS-101	C
30101	201409	2631	BUS-105	B
30102	201409	2631	ENG-101	B-

Figure 2: Grade Data File

From the grade file we can see that student 2630 took five courses in the Fall 2014 semester, receiving an "A-" in CS 100, an "S" (or Satisfactory) in FYS 101 (Freshman Experience), an "F" in MAT 112, a "B-" in NCS 181, and a "D+" in PHI 130. From the demographic file we discern that this student entered in Fall 2014, so these courses were taken in the first semester of the freshman year, and that this student entered with a high school average of 84, SAT Verbal of 640, and SAT Math of 750, and is a major in Network and Computer Security.

However, in order to bring the data together it was necessary to consult both files. The next step is to merge or **join** the files so that each record in the grade file also contains the demographic data elements.

Joining the Files

Joining the files is done in SPSS. Start SPSS and open both files in separate SPSS windows. SPSS can read Excel files, so opening these files should not be problematic. **Both files must be sorted in ascending order of ID**; if this is not the case it can be completed using **Data-Sort Cases** from the menu bar. To complete the join, go to the window containing the grades. Select **Data-Merge Files – Add Variables** as shown in Figure 3. It will be necessary to drag the ID variable (in this case **PSEUDOID**) from **Excluded Variables** to **Key Variables**.

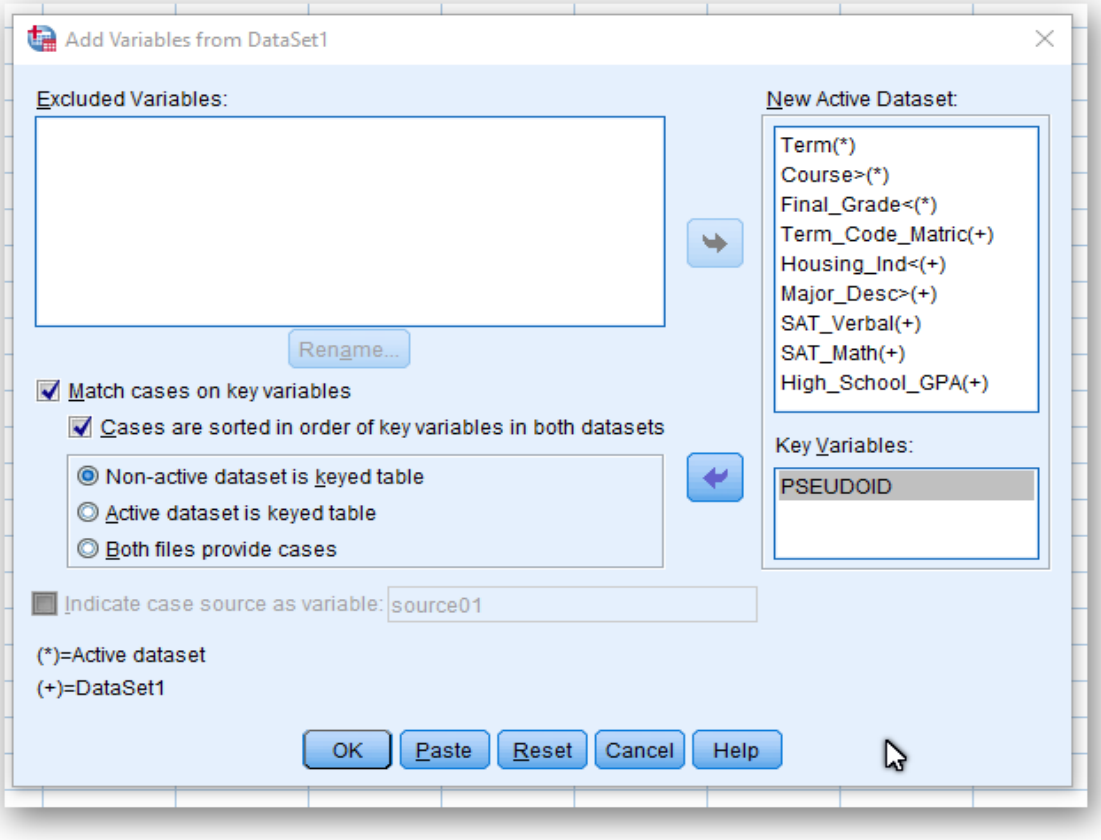

Figure 3: Joining the Files

The results of joining the files is shown in Figure 4.

	Term	PSEUDOID	Course	Final_Grade	Term_Code_Matric	Housing_Ind	Major_Desc	SAT_Verbal	SAT_Math	High_School_GPA
30069	201501	2629	CS-543	F	201401	N	Computer & Information Science	.	.	.
30070	201506	2629	CS-543	F	201401	N	Computer & Information Science	.	.	.
30071	201409	2630	CS-100	A-	201409	Y	Network and Computer Security	640	750	84.0000
30072	201409	2630	FYS-101	S	201409	Y	Network and Computer Security	640	750	84.0000
30073	201409	2630	MAT-112	F	201409	Y	Network and Computer Security	640	750	84.0000
30074	201409	2630	NCS-181	B-	201409	Y	Network and Computer Security	640	750	84.0000
30075	201409	2630	PHI-130	D+	201409	Y	Network and Computer Security	640	750	84.0000
30076	201501	2630	CS-108	W	201409	Y	Network and Computer Security	640	750	84.0000
30077	201501	2630	HIS-101	B	201409	Y	Network and Computer Security	640	750	84.0000
30078	201501	2630	IDS-103	C	201409	Y	Network and Computer Security	640	750	84.0000
30079	201501	2630	STA-100	F	201409	Y	Network and Computer Security	640	750	84.0000
30080	201509	2630	COM-106	B+	201409	Y	Network and Computer Security	640	750	84.0000
30081	201509	2630	MAT-112	A	201409	Y	Network and Computer Security	640	750	84.0000
30082	201509	2630	STA-100	C+	201409	Y	Network and Computer Security	640	750	84.0000
30083	201601	2630	CHI-101	A	201409	Y	Network and Computer Security	640	750	84.0000
30084	201601	2630	CS-108	B-	201409	Y	Network and Computer Security	640	750	84.0000
30085	201601	2630	MAT-115	B-	201409	Y	Network and Computer Security	640	750	84.0000
30086	201609	2630	NCS-210	W	201409	Y	Network and Computer Security	640	750	84.0000
30087	201609	2630	PHY-101L	W	201409	Y	Network and Computer Security	640	750	84.0000
30088	201609	2630	PHY-101T	W	201409	Y	Network and Computer Security	640	750	84.0000
30089	201609	2630	SOC-100	W	201409	Y	Network and Computer Security	640	750	84.0000
30090	201701	2630	COM-306	B+	201409	Y	Network and Computer Security	640	750	84.0000
30091	201701	2630	NCS-205	A	201409	Y	Network and Computer Security	640	750	84.0000
30092	201701	2630	PSY-100	B	201409	Y	Network and Computer Security	640	750	84.0000
30093	201709	2630	IS-310	D	201409	Y	Network and Computer Security	640	750	84.0000
30094	201709	2630	NCS-210	F	201409	Y	Network and Computer Security	640	750	84.0000
30095	201709	2630	PHY-101L	C+	201409	Y	Network and Computer Security	640	750	84.0000
30096	201709	2630	PHY-101T	F	201409	Y	Network and Computer Security	640	750	84.0000
30097	201801	2630	NCS-315	F	201409	Y	Network and Computer Security	640	750	84.0000
30098	201801	2630	NCS-320	A-	201409	Y	Network and Computer Security	640	750	84.0000
30099	201801	2630	NCS-330	D	201409	Y	Network and Computer Security	640	750	84.0000
30100	201409	2631	BUS-101	C	201409	N	Business Administration	370	400	86.2000
30101	201409	2631	BUS-105	B	201409	N	Business Administration	370	400	86.2000
30102	201409	2631	ENG-101	B-	201409	N	Business Administration	370	400	86.2000
30103	201409	2631	FYS-101	S	201409	N	Business Administration	370	400	86.2000

Figure 4: Results of the File Join

Notice that for each of the courses taken by student 2630, data for all variables in the demographic file have been added. Likewise, at the bottom of the figure, corresponding data for student 2631 has likewise been added.

At this point it is good practice to save the file. Should the user make a mistake at a later point in the process, recovery could start at this point.

Coding the Course Outcome

We have a course grade for each student; however, we are not attempting to build a model that predicts the actual course grade, but rather whether the course was satisfactorily completed. For our purposes at SUNY-Poly we have opted to consider a satisfactory outcome as a grade of "C" or higher; lower grades or a withdrawal are considered unsatisfactory. To achieve this it is necessary to construct a new variable. This is done in SPSS by selecting **Transform – Recode into Different Variables** as shown in Figure 5.

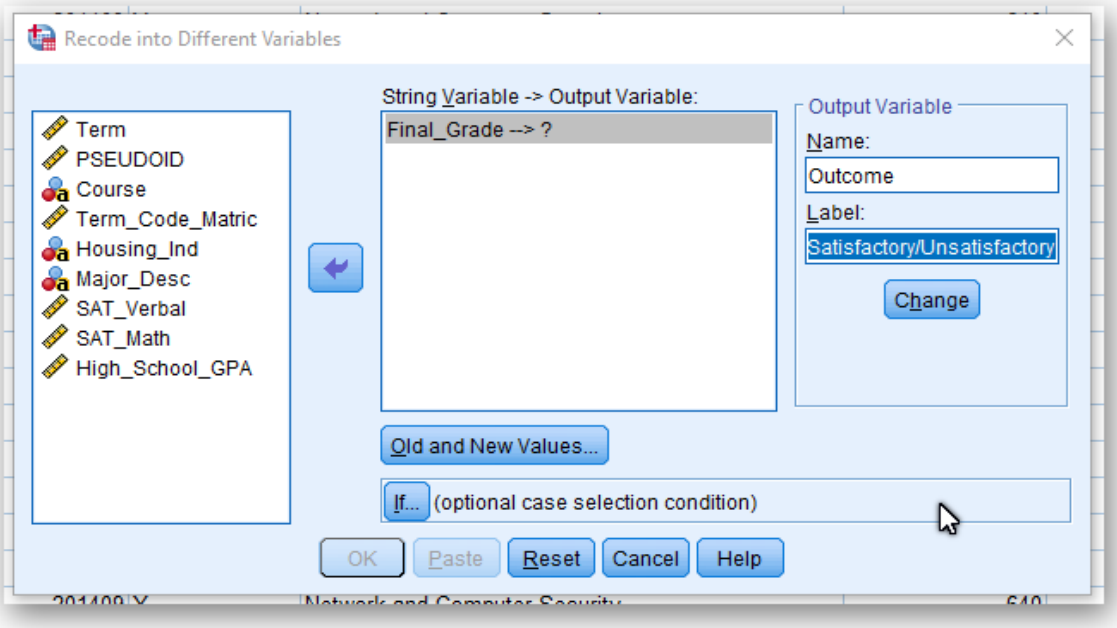

Figure 5: Recode into Different Variable

Drag the **Final_Grade** from the list of variables on the left into the **Output Variable** box in the center. Give the new variable a name (**Outcome** in this case) and a label. Click on the **Change** button (**very important)**. The name of the new variable will move into the center panel – See Figure 6. Then click on the **Old and New Values** button.

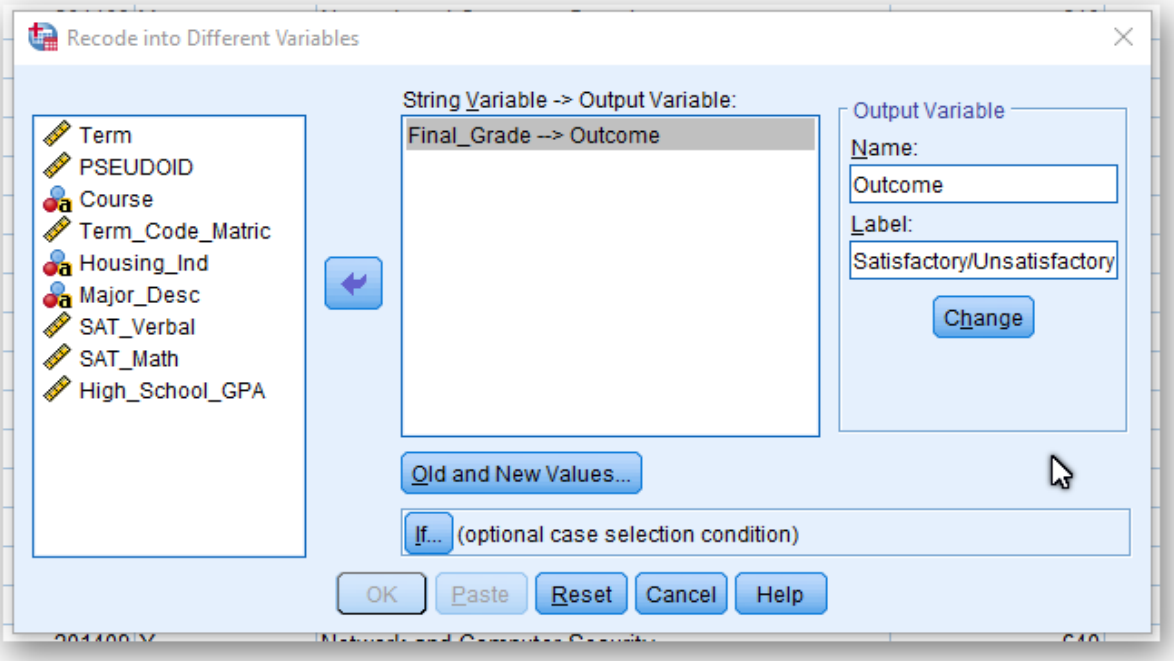

Figure 6: New Variable moved to Center Panel after clicking Change

The grades A+ through C, and an S grade (in S/U graded courses) have to be converted to a satisfactory outcome, denoted as S. All other grades have to be converted to an unsatisfactory outcome, denoted by U. Working one grade at a time, enter that grade into the **Old Value** field. The new variable is a string rather than a number, so the box **Output Variables are Strings** is checked, and the width set to one character (see Figure 7). In the example shown, an A+ will be converted to an S.

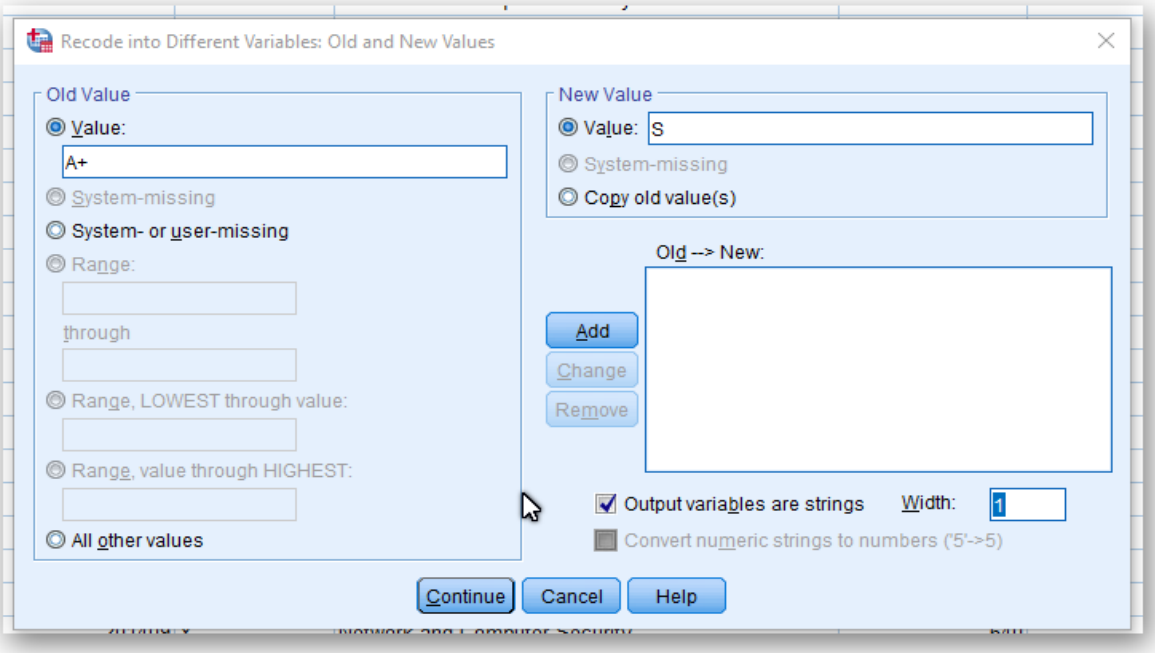

Figure 7: Grade Conversion

After each grade is so noted, the **Add** button must be clicked. The resulting conversion will be described in the **Old->New** pane.

Once all the satisfactory grades have been defined in this manner, all other grades are unsatisfactory. Click the **All other values** button on the left and enter **U** as the **New Value** on the right. Click the **Add** button. Results are shown in Figure 8.

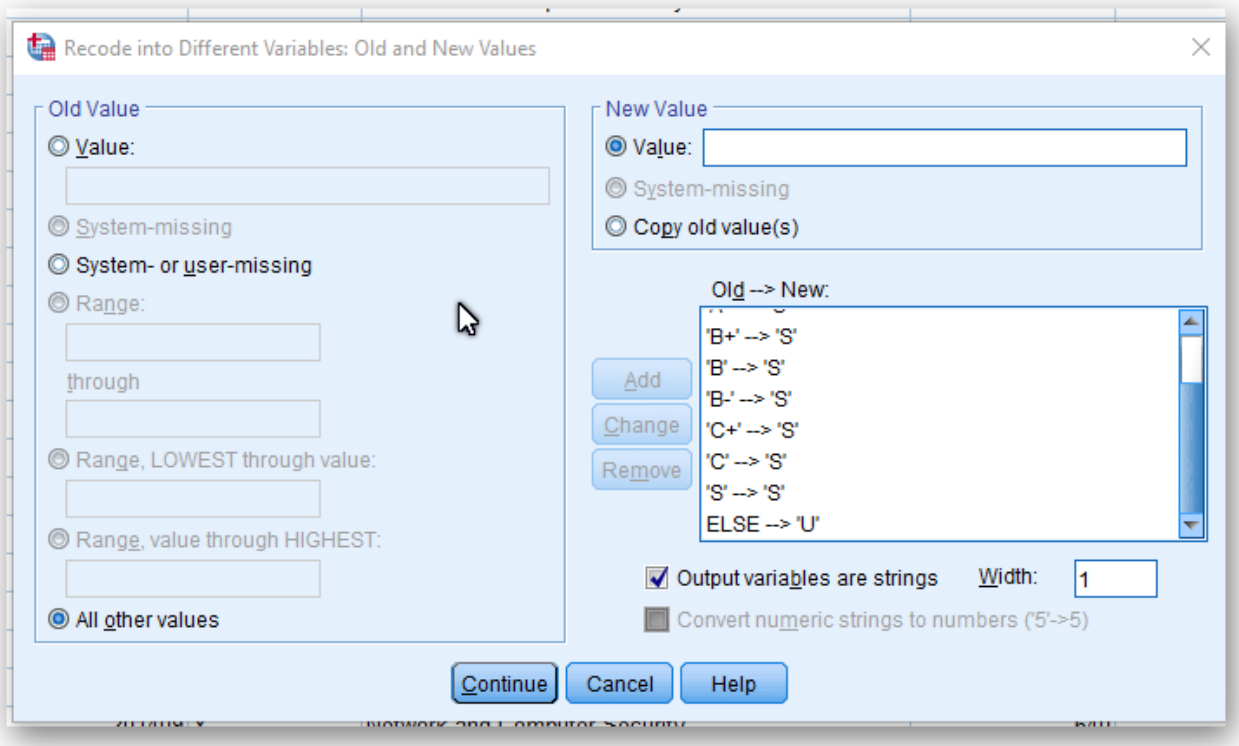

Figure 8: All Conversion Values Defined

Note that **S** is converted to an **S**. This is done because some courses (like FYS-101, Freshman Experience) are graded Satisfactory/Unsatisfactory, and Satisfactory is the desired outcome. If this rule were not included, and all other values were defined to be a **U**, the satisfactory outcome in these courses would be miscoded as unsatisfactory. When all rules have been defined as shown in Figure 8, click on **Continue** and then on **OK**. Results are shown in Figure 9. Note that the rightmost column now contains the new **Outcome** variable, and quickly perusing it reveals the correct calculation of this variable. Save the file at this point; it will become the working file.

	Term	PSEUDOID	Course	Final_Grade	Term_Code_Matric	Housing_Ind	Major_Desc	SAT_Verbal	SAT_Math	High_School_GPA	Outcome
30073	201409	2630	MAT-112	F	201409	Y	Network and Computer Security	640	750	84.0000	U
30074	201409	2630	NCS-181	B-	201409	Y	Network and Computer Security	640	750	84.0000	S
30075	201409	2630	PHI-130	D+	201409	Y	Network and Computer Security	640	750	84.0000	U
30076	201501	2630	CS-108	W	201409	Y	Network and Computer Security	640	750	84.0000	U
30077	201501	2630	HIS-101	B	201409	Y	Network and Computer Security	640	750	84.0000	S
30078	201501	2630	IDS-103	C	201409	Y	Network and Computer Security	640	750	84.0000	S
30079	201501	2630	STA-100	F	201409	Y	Network and Computer Security	640	750	84.0000	U
30080	201509	2630	COM-106	B+	201409	Y	Network and Computer Security	640	750	84.0000	S
30081	201509	2630	MAT-112	A	201409	Y	Network and Computer Security	640	750	84.0000	S
30082	201509	2630	STA-100	C+	201409	Y	Network and Computer Security	640	750	84.0000	S

Figure 9: Joined Data Set With Converted Grades

Course Selection

It is highly unlikely that models will be built for all courses offered; doing so is too labor intensive and including courses with limited enrollment (for example directed study courses offered for a handful of students) makes no sense. Assuming that the highest priority is to build models of courses typically taken by freshmen in their first semester of enrollment, it is necessary to identify those courses. Conventional wisdom may be a first approach, but it is really quite simple to let SPSS identify those courses.

We can start by assuming that where the semester in which the course is taken is the same as the semester the student matriculated, the course was taken in the first semester of the freshman year. While generally true, it may also be the case that where the two are the same, the student is a first-semester transfer. Nonetheless, this may be as close as we can get.

Start SPSS and load the working file. Click on **Data – Select Cases**. In the dialog box click on the **If condition is satisfied** radio button and click on **If**

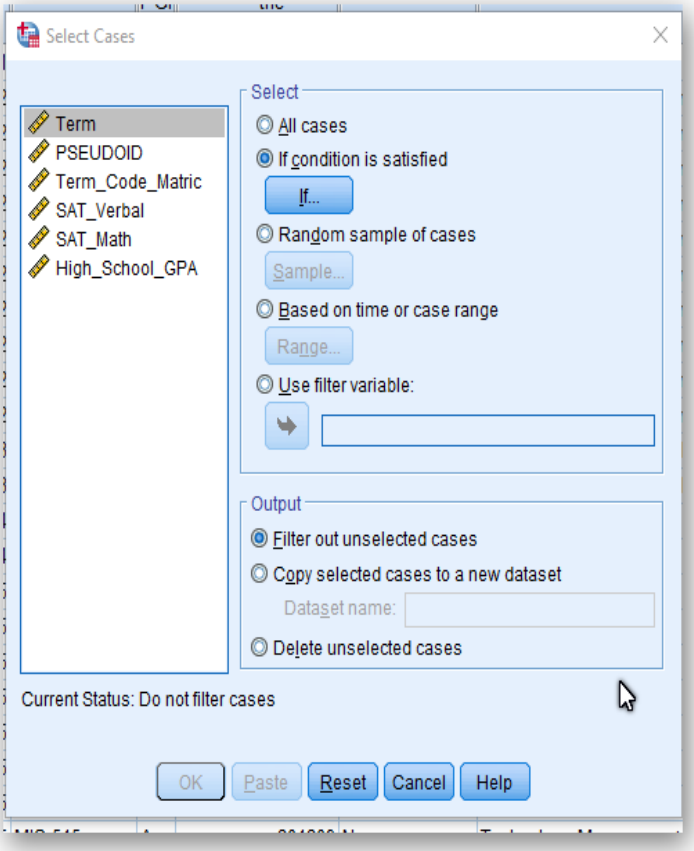

Figure 10: Select Cases Dialog

From the list of variables on the left, drag **Term** into the center area as shown, click on the = button, then drag **Term_Code_Matric** into the center as shown in Figure 11. When the formula is complete, click on the **Continue** button and then **OK** to activate the filter.

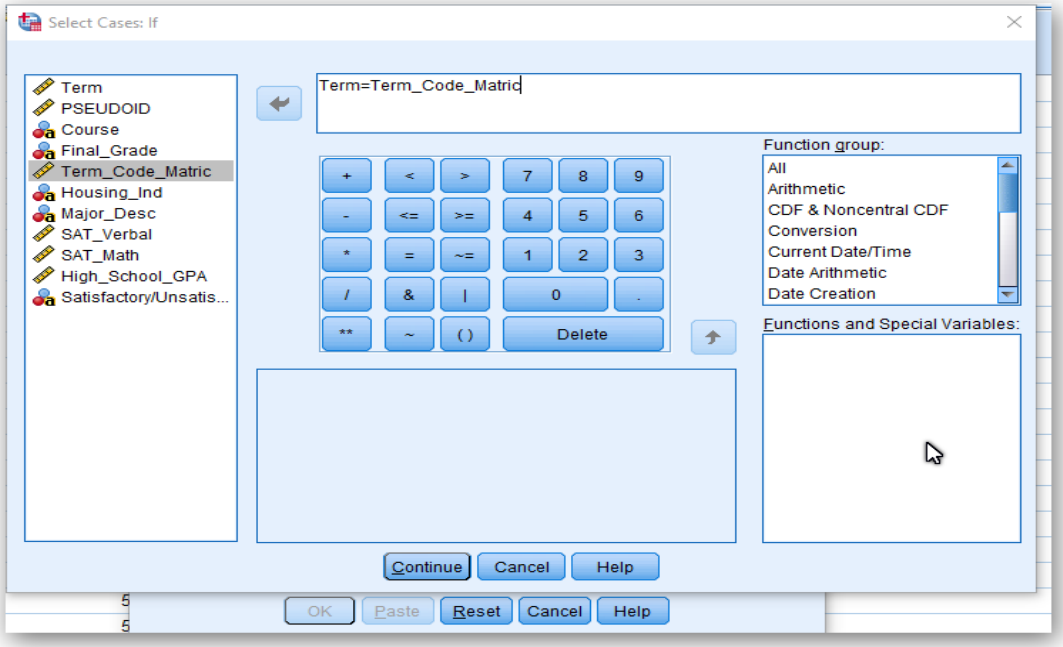

Figure 11: Entering the Selection Criteria

Results are displayed in Figure 12. Returning to our example student, ID 2630 who entered in Fall 2014, note that rows 30071 through 30075 have no diagonal lines through them. These lines met the filter; that is, the semester the course was taken is the same as the semester the student matriculated. Note that the lines beginning with row 30076 have a diagonal through the line numbers: they are excluded. Thus the rule has identified the correct cases.

	Term	PSEUDOID	Course	Final_Grade	Term_Code_Matric	Housing_Ind	Major_Desc	SAT_Verbal	SAT_Math	High_School_GPA	Outcome	filter_$
30071	201409	2630	CS-100	A-	201409	Y	Network and Computer Security	640	750	84.0000	S	1
30072	201409	2630	FYS-101	S	201409	Y	Network and Computer Security	640	750	84.0000	S	1
30073	201409	2630	MAT-112	F	201409	Y	Network and Computer Security	640	750	84.0000	U	1
30074	201409	2630	NCS-181	B-	201409	Y	Network and Computer Security	640	750	84.0000	S	1
30075	201409	2630	PHI-130	D+	201409	Y	Network and Computer Security	640	750	84.0000	U	1
30076	201501	2630	CS-108	W	201409	Y	Network and Computer Security	640	750	84.0000	U	0
30077	201501	2630	HIS-101	B	201409	Y	Network and Computer Security	640	750	84.0000	S	0
30078	201501	2630	IDS-103	C	201409	Y	Network and Computer Security	640	750	84.0000	S	0
30079	201501	2630	STA-100	F	201409	Y	Network and Computer Security	640	750	84.0000	U	0
30080	201509	2630	COM-106	B+	201409	Y	Network and Computer Security	640	750	84.0000	S	0
30081	201509	2630	MAT-112	A	201409	Y	Network and Computer Security	640	750	84.0000	S	0

Figure 12: Results of Applying the Filter

Next, do a frequency count of course number. Select **Analyze-Descriptive Statistics-Frequencies** (first select **Analyze**, then from the menu that emerges **Descriptive Statistics** and so on) and drag **Course** into the variable list as shown in Figure 13.

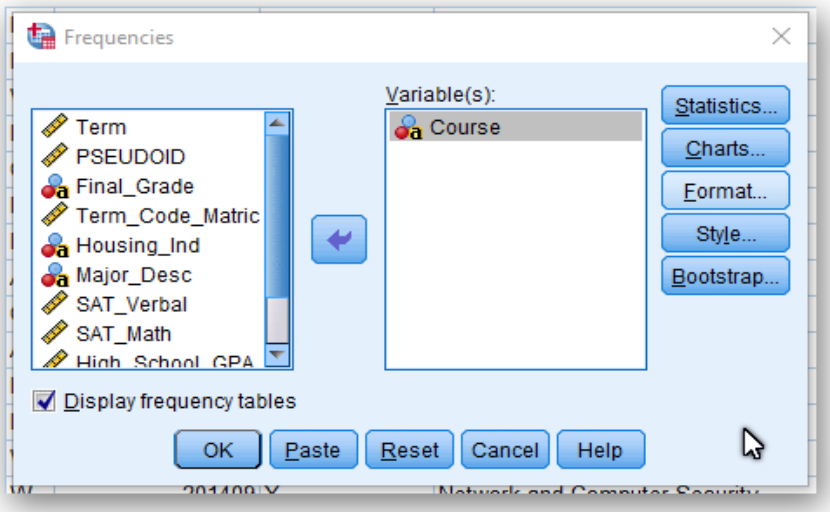

Figure 13: Frequencies Dialog

Next, Click on the **Format** button and select **Descending Counts** as shown in Figure 14. Click on **Continue** and then **OK** to activate.

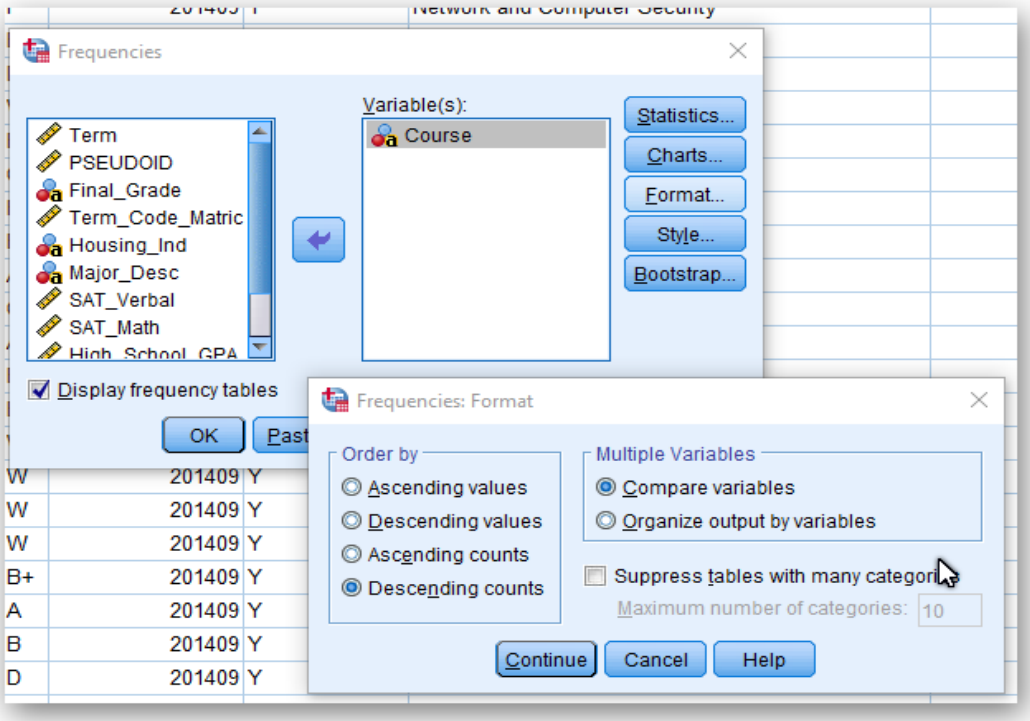

Figure 14: Arranging Courses in Descending Order of Enrollment

Results are displayed in the **Output Window** as shown in Figure 15. Note that the total enrollment for courses taken in the first semester of enrollment is 20,555. The single most frequently enrolled course was FYS-101, the Freshman Experience course. This is hardly surprising since almost all first-year students take that course in their first semester. This course accounted for 7.7% of all first- semester course enrollments over this five-year period. However, since it is a one-credit course that students are almost guaranteed to pass if they attend, no model will be built for this course.

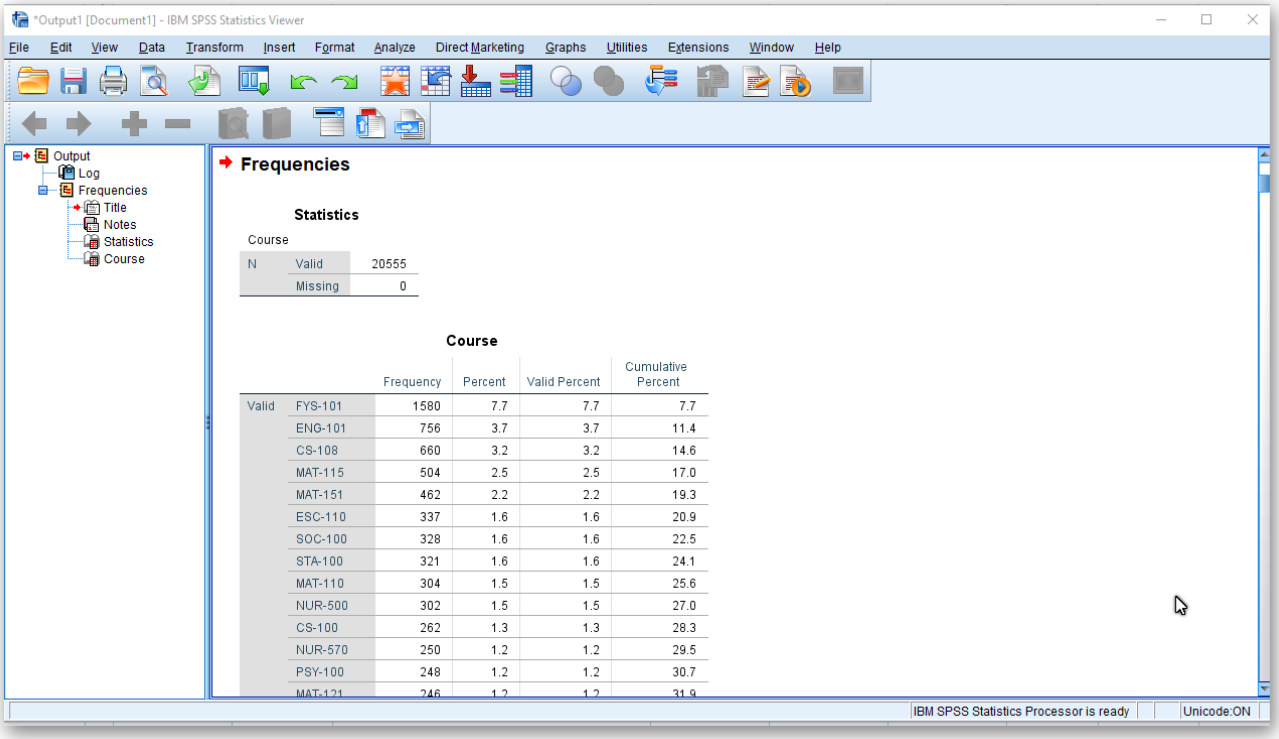

Figure 15: First Semester Course Enrollment in Descending Order of Enrollment

From this table we discern that the five most heavily enrolled courses taken in the first semester of the first year are:

- ENG 101 – Freshman English
- CS 108 – Computing Fundamentals
- MAT 115 – Finite Mathematics
- MAT 151 – Calculus I
- ESC 110 – Introduction to Engineering

For this illustration, a model will be built for ENG-101, again based solely upon the performance of students who took it in their first semester of enrollment; that is, based on the performance of the 756 students identified in Figure 15.

Building the Model

The first task is to select the correct cases: to select only the 756 students who took the course in their first semester of enrollment. Figure 15 tells us that there are 756, but all other cases have to be filtered out. This is again accomplished by using **Data-Select Cases** as shown in Figure 16.

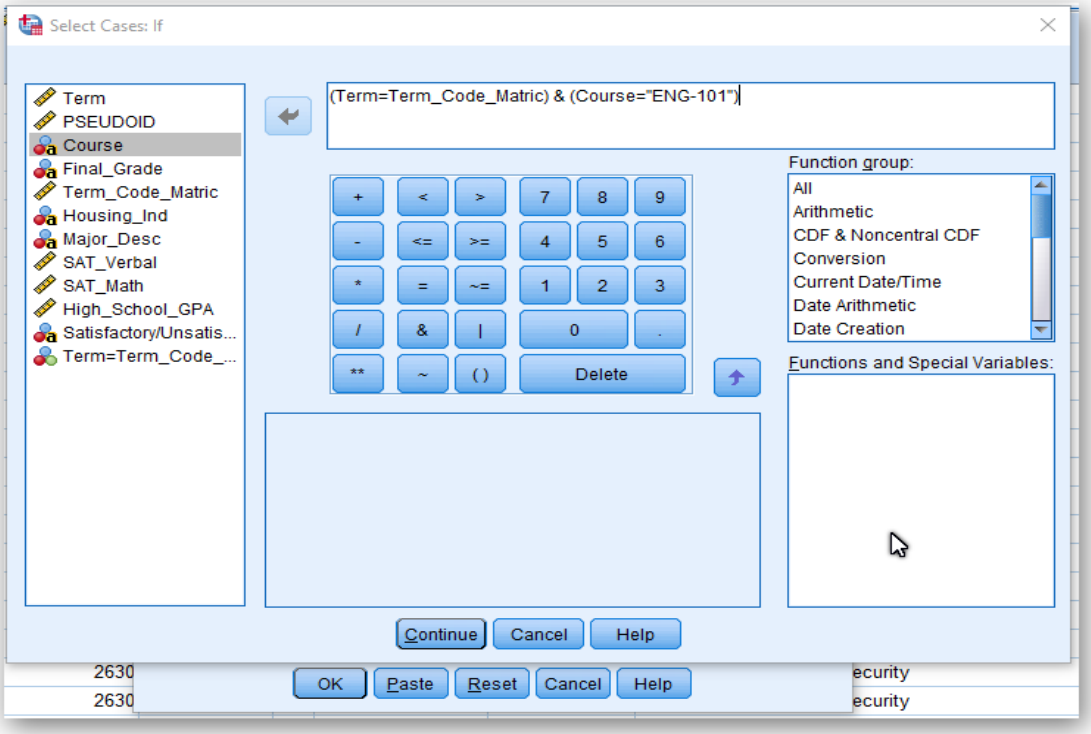

Figure 16: Selecting Only Cases where ENG 101 was taken in the first semester of enrollment

Carefully examine the left expression (Term=Term_Code_Matric). As before this ensures that only cases where the course was taken in the first semester of enrollment is included. The right expression ensures that only cases where the course is ENG-101 are included. Note that the course name is a string (a series of characters). and the string must be encapsulated within double quote marks. Finally, the **&** symbol (and) ensures that only cases meeting both the left expression and the right expression are included, so only cases where the course was taken in the first semester **and** the course is ENG-101. When this is complete, re-run the frequency count shown in Figures 13 and 14; the result should be that there are 756 cases and all are enrolled in ENG-101. This result is shown in Figure 17.

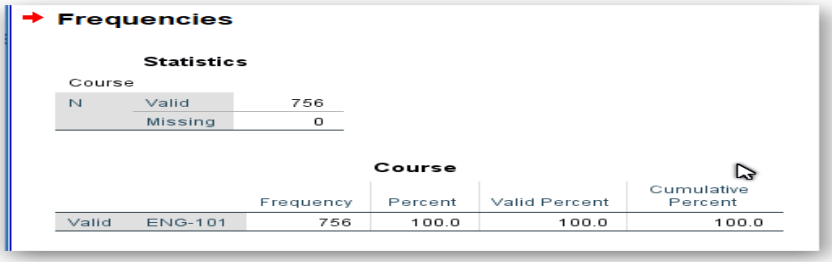

Figure 17: Results of the Course and Semester Filter

Since the results here match expectations, we can proceed, confident that the correct cases have been selected – that we have identified all cases where ENG 101 was taken in the first semester of enrollment.

To build the model, select **Analyze-Classify-Tree** from the menu bar. Click on **OK** to bring up the **Decision Tree** dialog window. Drag the variable we are trying to predict – course performance **(Satisfactory/Unsatisfactory)** into the **Dependent Variable** area. Drag the variables that **might** be used to make the prediction into the **Independent Variables** area as shown. Thus, in this example, an attempt will be made to predict course outcome on the basis of high school average, SAT scores, major, and whether the student lives on or off campus.

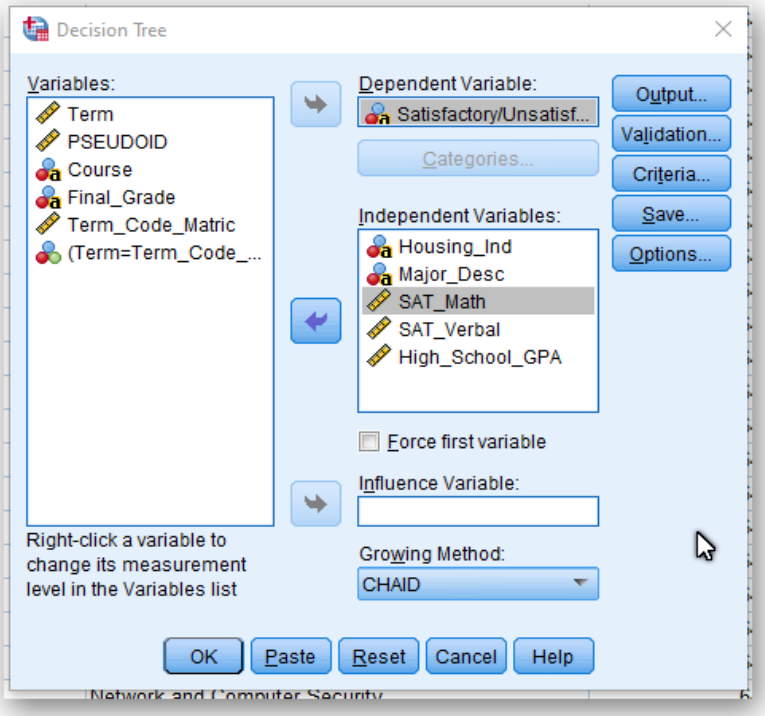

Figure 18: Decision Tree Dialog

Next, click on the **Criteria** button. In the ensuing dialog the minimum number of cases in a node in the tree is specified. The default is 50 cases in a node, and 100 cases in the parent node. Because SUNY Poly is a small institution, requiring 50 cases is unrealistic. In Figure 19, the minimum number of cases is adjusted to 5 and 10 respectively. At a large campus the default values can be accepted; at a small campus they will need to be adjusted. Click on **Continue** and then **OK.**

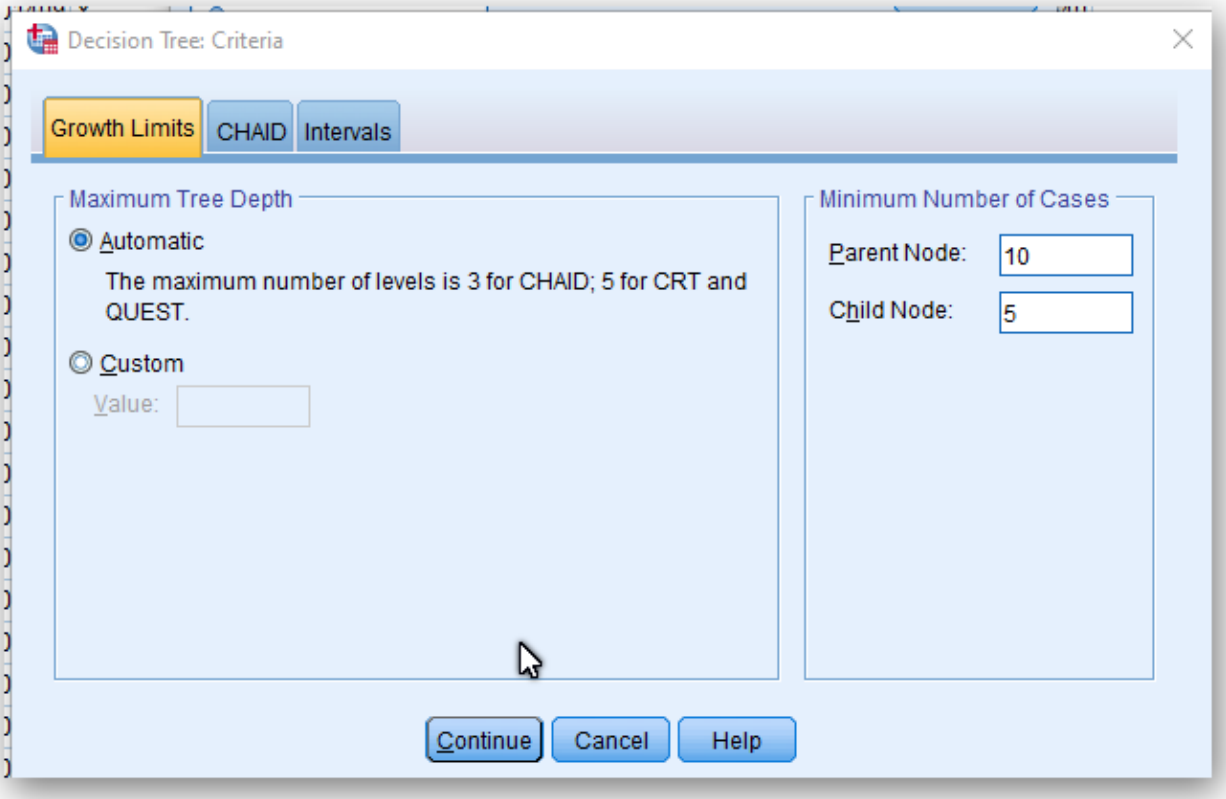

Figure 19: Adjusting the Minimum Number of Cases

The decision tree is built, and the results are shown in Figure 20. Of the 756 students who took ENG-101 in their first semester, 652 or 86.2% had a satisfactory outcome (a grade of "C" or higher), and 104 or 13.8% did not. The tree shows that using this algorithm (CHAID) the **only** predictive variable is high school average. Not a single student with a high school average greater than 94.85 had an adverse outcome over this five-year period and there were 73 such students. For students with a high school average between 88.66 and 94.85 (or no high school average in the database), 23 out of 313 or 7.3% had an adverse outcome. Among the 296 students who entered with a high school average between 82.99 and 88.65, 55 or 18.6% had an adverse outcome, and for those with a high school average below 82.99 26 of 74 or 35.1% had an adverse outcome.

What is particularly interesting here is that all students who took ENG 101 in their freshman year passed a placement exam, yet for those entering with a low high school average, ENG 101 still presented a substantial risk.

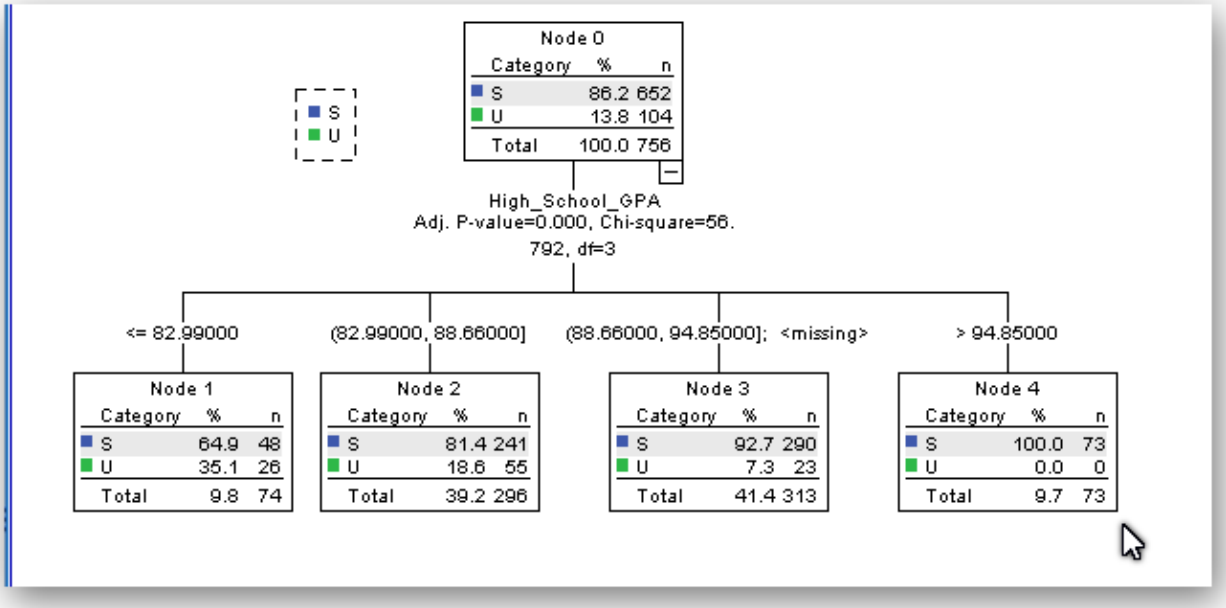

Figure 20: Decision Tree for ENG 101 based on CHAID Algorithm

SPSS Decision Trees provides four different algorithms that can be used for tree generation. As a general rule it is a good idea to try all of the algorithms to see if an alternate provides easier, more understandable rules or is more discerning. Alternate algorithms can be selected in the **Growing Method** dropdown shown in Figure 18.

In this example, when the second algorithm, **Exhaustive CHAID** is used, the exact same results occur and are not shown here.

Using the CRT method generates a tree of up to five levels. For readability, only the top three levels of the tree are displayed in Figure 21. Once again, high school average is the best predictor. Among students with a high school average higher than 86.795, only 7.7% had an adverse outcome; among those with a lower high school average 26.3% had an adverse outcome. The third level of the tree further breaks down those with a high school average of 86.795 or less. Within this group, students majoring in one of a long list of majors (See Node 3) had more than one-third with an unsatisfactory outcome, while another list of majors with the same high school average profile had only 16.8% with an unsatisfactory outcome. If we use a rule of thumb that students in a group where 30% or more have an unsatisfactory outcome are at high risk, then the model using CHAID identified 26 of the 104 unsatisfactory cases (See Figure 20, Node 1), whereas the CRT model identified 47 of the 104. Thus, the CRT model is probably a more discerning one.

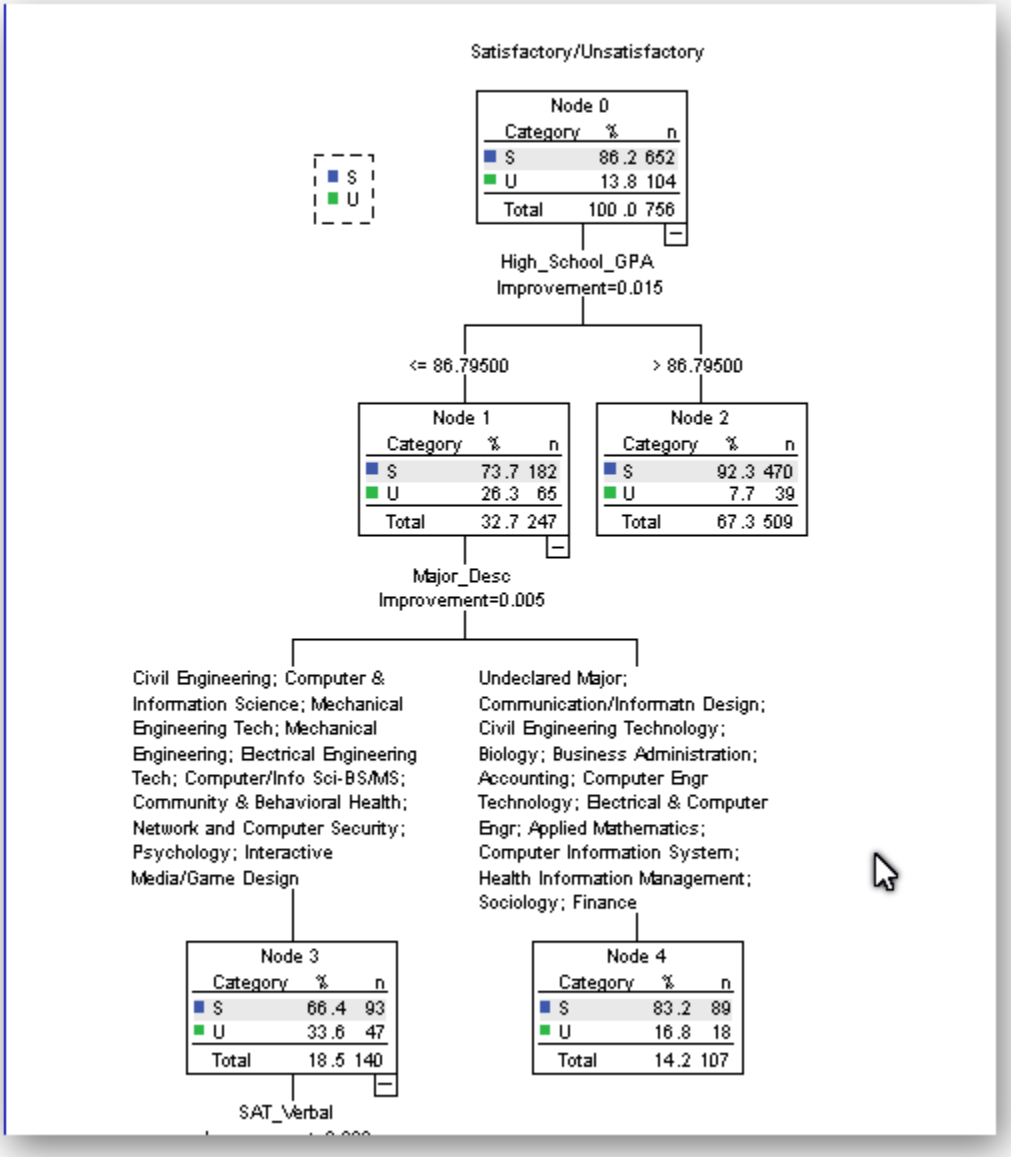

Figure 21: Decision Tree for ENG 101 based on CRT algorithm (bottom of tree amputated to facilitate easy display)

The final algorithm (**Quest**) produced a tree that only identified four cases where an unsatisfactory outcome exceeded 30% and is clearly inferior. It is not shown here.

At SUNY-Poly students falling into Node 3 are deemed high-risk in this course. Entering first-year students are pre-registered for courses. Several of our academic programs use this riskanalysis when making course selections, others do not. In programs that do, a conscious attempt is made to minimize or eliminate high-risk choices, especially during the critical first semester of the first year. Thus, where possible, a student falling into Node 3 would not be scheduled for ENG 101 during the first semester of the first year; the student will have to take the course, but hopefully will do so in a semester when not enrolled in any other high-risk courses.

It is important to remember that these models are probabilistic, not deterministic. As such, they are no different than placement exams:a good score on a placement exam does not guarantee that the student will succeed in the target course, and a poor score does not guarantee that the student will not perform well. Decision trees will rarely find combinations of characteristics where an unsatisfactory outcome is more likely than a satisfactory outcome. Thus, interpretation of the results becomes important. In a course where, on average, 15% of the enrolled students do not perform well, a student with a 30% probability of not doing well is at substantial risk. Even at that higher risk level the student is more likely to succeed than to fail, but in this example the student is twice as likely as the typical student in that course to perform poorly. Information is power, but it needs to be used wisely.

As educators we want to challenge our students and to encourage our students to challenge themselves, but we also should not put them into situations where they are unlikely to succeed. What is needed is a balance. Placing a student into one challenging course at a time may represent a reasonable risk, but is placing that same student into three or four challenging courses in the same semester, particularly the first semester of the first year in college also reasonable? Some academic programs are very flexible, and it is easy for advisors and students to find that balance. Other academic programs are highly structured, with few choices. However, it is rare that programs have no choices. An engineering major, for example, may be required to take a history course, but there is still a choice as to which one. Faced with challenging math and science courses in the freshman year, is it possible to identify a history course where the student is not at risk? Again, here at SUNY-Poly we have identified groups of students for whom American History I (pre-Civil War) represents a high risk, whereas American History II (post-Civil War) does not, and the exact opposite holds for other identifiable groups. If the student is going to take an American History course in any event, is it not reasonable to counsel this engineering major into the American History course that does not represent a high risk?

Conclusion

Developing advisement protocols based upon actual course enrollment history is not difficult. The steps involved in producing decision-tree models of course performance is readily accomplished using standard computing tools that are typically found on college campuses. The “knowledge” extracted from this type of exercise is likely to substantially exceed the knowledge base of even the most experienced academic advisors. Even if an academic advisor “knows” that a particular variable is a good outcome predictor, that individual is highly unlikely to know the “cut point” separating high-risk from moderate-risk – and, moreover, that cut point is likely to change over time. Data based decision models are sensitive to those changes and are readily adapted.