Knowledge Discovery from Data as a Tool for Determining Successful Placement of Students in Their First College Math Course

Kelle Clark, CSCI6840 Spring 2020
April 26, 2020

Abstract

It is crucial to the success of a first-year college student that they are provided with the best guidance in course selection. Wingate University has dedicated a lot of effort in creating a Math Placement Test under the direction of Dr. Lauer, Dr. Joyner and Dr. Bell of the Wingate Math Department. The math faculty use the results of a student's Math Placement Test, along with other data, to recommend a list of math courses suitable for the student's skill level. The results presented can be used in measuring the effectiveness of this advising process highlighting the methods of data mining in classification of students and identifying potential patterns in the data that were not yet detected.

Section 1 Introduction

1.1. The Conditions Surrounding the Data Collection

Wingate University began as Wingate School almost 125 years ago founded in response to the need for an institution that could offer a literacy education to those in the Carolina Piedmont area. The school, situated on a serene plot of 10 acres with an enrollment of approximately 175 students in grades 1 through 12, had, by the fall of 2018, become the fastest growing independent college or university in North Carolina. With a 37\% growth in student population from 2013 to 2018 and a continuing upward trend in enrollment, each year brings Wingate closer to becoming a 4,000 to 6,000 student University. Rhett Brown, President of Wingate University, points out that the institution's growth is rooted with purpose. The Wingate 2017-2018 Impact Report says it best: "...how many people's lives are changed for the better because of the knowledge and experiences (a Wingate graduate has) gained.." Wingate University, consistently ranked among the top 20 "best value" Universities in the South by U.S. News and World Report, is meeting the need of the steadily increasing number of people seeking programs designed to improve student success in a challenging economy. The careful construction of the academic advising process that in part attempts to place students in math courses most likely to result in the positive outcome of "pass."

The growth of WU can be attributed to the establishment of two satellite campuses, extended graduate program offerings in Education and Pharmacy and PA professional programs, and the impetus behind this paper, "tweaking admissions for undergraduates." The Academic year 2017-2018 was a pivotal year for Wingate University with a jump of first-year undergraduate students increasing the previous year's number by 62 percent. The Mathematics Department had prepared the year before for the new influx of students and the challenge of accurately guiding this incoming class of students with the development of a Math Placement Test. The Math

Placement test was first offered to the incoming freshman class of 2017. The use of the test results was originally intended to improve the accuracy of prediction based on standardized test scores and HSGPA (high school grade point average) in the success of a student in each of the typical first math courses available at Wingate: College Algebra (112), Quantitative Reasoning (116), PreCalculus (115), Inferential Statistics (209), Calculus for Business Majors (117), Calculus and Analytic Geometry (120).

In the sections that follow we review what information has been collected (section 1.2), summarize the conclusions made after the first two years of offering the Math Placement Test (section 1.3), demonstrate the techniques the author has learned in CSCI 6840 Data Mining (section 2.1), draw conclusion from the research (section 3.1) and list possible directions for future research (section 3.2).

1.2 The Collection Tool and Other Measurements

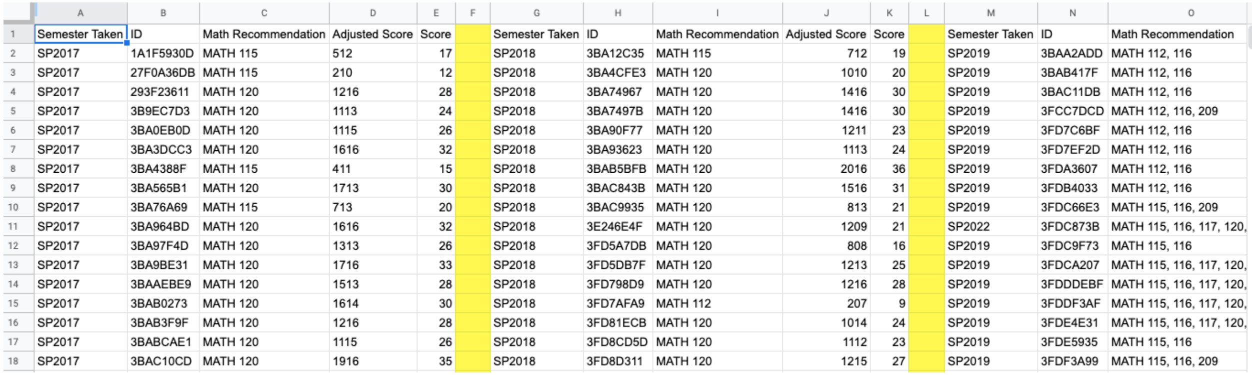
Every student graduating from Wingate University must take at least one math course. The application process for acceptance to Wingate asks a student to self-report their HSGPA and standardized test scores, with official reporting due by enrollment. Traditionally, the Math SAT scores are primarily used to place incoming students into math classes classification done by hand by one faculty member. This process was long, even when there were only 500 incoming students. As mentioned before, the number of incoming students has been steadily increasing making the desire to automate this process very desirable.

When a student has scored at least a 500 on the Math SAT or at least a 20 on the ACT, then there is strong empirical evidence that the student can take any course from MATH 115 and higher with a high probability of success. It is when the student's standard scores are below 500 or are not available that the process is not well defined. The distribution of student SAT score is roughly symmetric with a quarter of accepted students scoring below 500 on the Math SAT. Also, we will see later in section …… that, because Wingate has a rolling admission policy, not all students have finalized their application before the "batch" recommendation process must be done so that freshman advisor can prepare to meet with their students.

The Spring 2020 Corona Virus pandemic has created a new situation that will have an impact on the admission and recommendation process for many students and colleges/universities. With schools having to use remote learning options and seniors unable to take their AP exams and to take the SAT/ACT during the Spring of 2020, many schools are waiving the SAT and ACT requirements for the academic year 2020-2021. Wingate has not made that decision as yet, but the admission team is working diligently with students to guide them during this time. Flexibility is going to be the rule for all. But, long before this unfortunate circumstance, schools were beginning to weigh the options beyond standardized test scores for college admissions.

In preparation for the growing number of incoming students, in 2015 the Wingate Math Department set out to devise a Placement Test. The "beta" version of this test consists of 36 question as a Google Form. This tool was constructed to measure a student's procedural knowledge of basic algebra concepts and logical thinking. The data is directed into a Google sheet, and the information shares a common key of the student's Wingate id number with other database records containing the student application materials. The data available from the first year of implementation, Fall 2017 and the Falls since are in the Kelle Clark CSCI6840 Project/Original Data/Math Placement Results - - 2017 - 2019. A snippet of the Math Placement Test data from these three years as shared with me from the Math Department as a Google Sheet is provide below in Fig. 1.2.1.

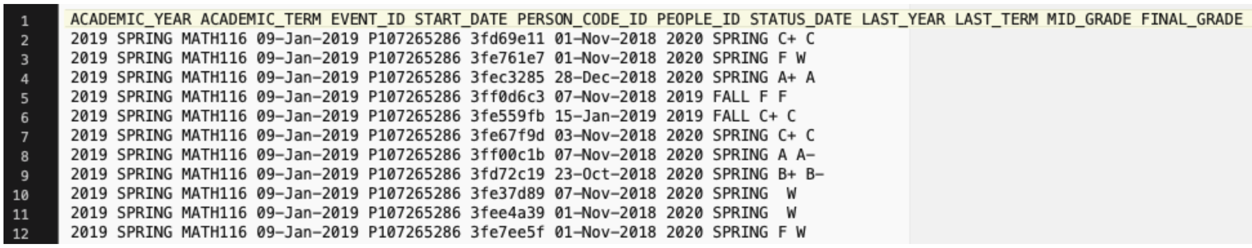
Fig 1.2.1



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Semester Taken	ID	Math Recommendation	Adjusted Score	Score		Semester Taken	ID	Math Recommendation	Adjusted Score	Score		Semester Taken	ID	Math Recommendation
2	SP2017	1A1F5930D	MATH 115	512	17		SP2018	3BA12C35	MATH 115	712	19		SP2019	3BAA2ADD	MATH 112, 116
3	SP2017	27F0A36DB	MATH 115	210	12		SP2018	3BA4CFE3	MATH 120	1010	20		SP2019	3BAB417F	MATH 112, 116
4	SP2017	293F23611	MATH 120	1216	28		SP2018	3BA74967	MATH 120	1416	30		SP2019	3BAC11DB	MATH 112, 116
5	SP2017	3B9EC7D3	MATH 120	1113	24		SP2018	3BA7497B	MATH 120	1416	30		SP2019	3FCC7DCD	MATH 112, 116, 209
6	SP2017	3BA0EB0D	MATH 120	1115	26		SP2018	3BA90F77	MATH 120	1211	23		SP2019	3FD7C6BF	MATH 112, 116
7	SP2017	3BA3DCC3	MATH 120	1616	32		SP2018	3BA93623	MATH 120	1113	24		SP2019	3FD7EF2D	MATH 112, 116
8	SP2017	3BA4388F	MATH 115	411	15		SP2018	3BAB5BFB	MATH 120	2016	36		SP2019	3FDA3607	MATH 112, 116
9	SP2017	3BA565B1	MATH 120	1713	30		SP2018	3BAC843B	MATH 120	1516	31		SP2019	3FDB4033	MATH 112, 116
10	SP2017	3BA76A69	MATH 115	713	20		SP2018	3BAC9935	MATH 120	813	21		SP2019	3FDC66E3	MATH 115, 116, 209
11	SP2017	3BA964BD	MATH 120	1616	32		SP2018	3E246E4F	MATH 120	1209	21		SP2022	3FDC873B	MATH 115, 116, 117, 120,
12	SP2017	3BA97F4D	MATH 120	1313	26		SP2018	3FD5A7DB	MATH 120	808	16		SP2019	3FDC9F73	MATH 115, 116
13	SP2017	3BA9BE31	MATH 120	1716	33		SP2018	3FD5DB7F	MATH 120	1213	25		SP2019	3FDCA207	MATH 115, 116, 117, 120,
14	SP2017	3BAAEBE9	MATH 120	1513	28		SP2018	3FD798D9	MATH 120	1216	28		SP2019	3FDDDEBF	MATH 115, 116, 117, 120,
15	SP2017	3BAB0273	MATH 120	1614	30		SP2018	3FD7AFA9	MATH 112	207	9		SP2019	3FDDF3AF	MATH 115, 116, 117, 120,
16	SP2017	3BAB3F9F	MATH 120	1216	28		SP2018	3FD81ECB	MATH 120	1014	24		SP2019	3FDE4E31	MATH 115, 116, 117, 120,
17	SP2017	3BABCAE1	MATH 120	1115	26		SP2018	3FD8CD5D	MATH 120	1112	23		SP2019	3FDE5935	MATH 115, 116
18	SP2017	3BAC10CD	MATH 120	1916	35		SP2018	3FD8D311	MATH 120	1215	27		SP2019	3FDF3A99	MATH 115, 116, 209

The pieces of the students' college records were accessed through the Registrar's office and shared with me as a .txt file and a .csv file. The student id's were removed and the common key value between these datasets was replaced with a hash function value and appears as ID in the table above and as PEOPLE_ID. Some normalization was done to case fold the key so that I could combine the two records for the same student using =VLOOKUP(key, table range list, column, True/False) excel formula.

Fig 1.2.2



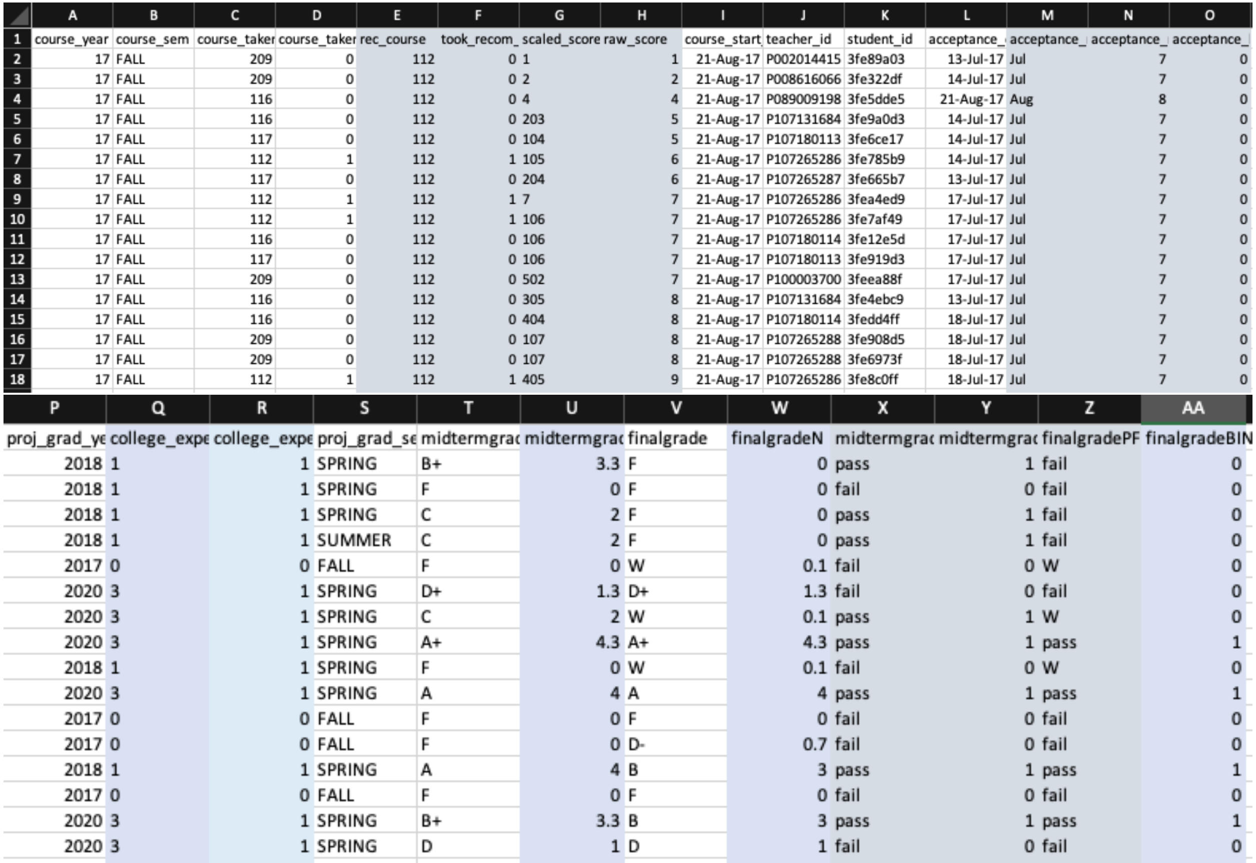
```
ACADEMIC_YEAR ACADEMIC_TERM EVENT_ID START_DATE PERSON_CODE_ID PEOPLE_ID STATUS_DATE LAST_YEAR LAST_TERM MID_GRADE FINAL_GRADE
2019 SPRING MATH116 09-Jan-2019 P107265286 3fd69e11 01-Nov-2018 2020 SPRING C+ C
2019 SPRING MATH116 09-Jan-2019 P107265286 3fe761e7 01-Nov-2018 2020 SPRING F W
2019 SPRING MATH116 09-Jan-2019 P107265286 3fec3285 28-Dec-2018 2020 SPRING A+ A
2019 SPRING MATH116 09-Jan-2019 P107265286 3ff0d6c3 07-Nov-2018 2019 FALL F F
2019 SPRING MATH116 09-Jan-2019 P107265286 3fe559fb 15-Jan-2019 2019 FALL C+ C
2019 SPRING MATH116 09-Jan-2019 P107265286 3fe67f9d 03-Nov-2018 2020 SPRING C+ C
2019 SPRING MATH116 09-Jan-2019 P107265286 3ff00c1b 07-Nov-2018 2020 SPRING A A-
2019 SPRING MATH116 09-Jan-2019 P107265286 3fd72c19 23-Oct-2018 2020 SPRING B+ B-
2019 SPRING MATH116 09-Jan-2019 P107265286 3fe37d89 07-Nov-2018 2020 SPRING  W
2019 SPRING MATH116 09-Jan-2019 P107265286 3fee4a39 01-Nov-2018 2020 SPRING  W
2019 SPRING MATH116 09-Jan-2019 P107265286 3fe7ee5f 01-Nov-2018 2020 SPRING F W
```

When creating the combined record, I followed the flow:

1. Case fold "ID" to all lower case
2. Rename "ID" and "PEOPLE_ID" to student_id
3. Combine sheets using the vlookup() formula with student_id as the shared key.

The final 2017 data was separated into three .csv and .xlsx files {MathPlacement2017extended, NoMathPlacement2017extended and Combined2017extended}. There were some missing values, such as a few midterm grades and a few final grades (especially for the year 2019). The author created new attributes from the given values using transformations/ linear combinations/ indirect values = results of conditionals. These attributes are indicated by the shaded columns.

Fig 1.2.3



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	course_year	course_sem	course_taker	course_taker	rec_course	took_recom_	scaled_score	raw_score	course_start	teacher_id	student_id	acceptance_	acceptance_	acceptance_	acceptance_
2	17	FALL	209	0	112	0	1	1	21-Aug-17	P002014415	3fe89a03	13-Jul-17	Jul	7	0
3	17	FALL	209	0	112	0	2	2	21-Aug-17	P008616066	3fe322df	14-Jul-17	Jul	7	0
4	17	FALL	116	0	112	0	4	4	21-Aug-17	P089009198	3fe5dde5	21-Aug-17	Aug	8	0
5	17	FALL	116	0	112	0	203	5	21-Aug-17	P107131684	3fe9a0d3	14-Jul-17	Jul	7	0
6	17	FALL	117	0	112	0	104	5	21-Aug-17	P107180113	3fe6ce17	14-Jul-17	Jul	7	0
7	17	FALL	112	1	112	1	105	6	21-Aug-17	P107265286	3fe785b9	14-Jul-17	Jul	7	0
8	17	FALL	117	0	112	0	204	6	21-Aug-17	P107265287	3fe665b7	13-Jul-17	Jul	7	0
9	17	FALL	112	1	112	1	7	7	21-Aug-17	P107265286	3fea4ed9	17-Jul-17	Jul	7	0
10	17	FALL	112	1	112	1	106	7	21-Aug-17	P107265286	3fe7af49	17-Jul-17	Jul	7	0
11	17	FALL	116	0	112	0	106	7	21-Aug-17	P107180114	3fe12e5d	17-Jul-17	Jul	7	0
12	17	FALL	117	0	112	0	106	7	21-Aug-17	P107180113	3fe919d3	17-Jul-17	Jul	7	0
13	17	FALL	209	0	112	0	502	7	21-Aug-17	P100003700	3feea88f	17-Jul-17	Jul	7	0
14	17	FALL	116	0	112	0	305	8	21-Aug-17	P107131684	3fe4ebc9	13-Jul-17	Jul	7	0
15	17	FALL	116	0	112	0	404	8	21-Aug-17	P107180114	3fedd4ff	18-Jul-17	Jul	7	0
16	17	FALL	209	0	112	0	107	8	21-Aug-17	P107265288	3fe908d5	18-Jul-17	Jul	7	0
17	17	FALL	209	0	112	0	107	8	21-Aug-17	P107265288	3fe6973f	18-Jul-17	Jul	7	0
18	17	FALL	112	1	112	1	405	9	21-Aug-17	P107265286	3fe8c0ff	18-Jul-17	Jul	7	0

P	Q	R	S	T	U	V	W	X	Y	Z	AA
proj_grad_ye	college_expe	college_expe	proj_grad_se	midtermgrac	midtermgrac	finalgrade	finalgradeN	midtermgrac	midtermgrac	finalgradePF	finalgradeBIN
2018	1	1	SPRING	B+	3.3	F	0	pass	1	fail	0
2018	1	1	SPRING	F	0	F	0	fail	0	fail	0
2018	1	1	SPRING	C	2	F	0	pass	1	fail	0
2018	1	1	SUMMER	C	2	F	0	pass	1	fail	0
2017	0	0	FALL	F	0	W	0.1	fail	0	W	0
2020	3	1	SPRING	D+	1.3	D+	1.3	fail	0	fail	0
2020	3	1	SPRING	C	2	W	0.1	pass	1	W	0
2020	3	1	SPRING	A+	4.3	A+	4.3	pass	1	pass	1
2018	1	1	SPRING	F	0	W	0.1	fail	0	W	0
2020	3	1	SPRING	A	4	A	4	pass	1	pass	1
2017	0	0	FALL	F	0	F	0	fail	0	fail	0
2017	0	0	FALL	F	0	D-	0.7	fail	0	fail	0
2018	1	1	SPRING	A	4	B	3	pass	1	pass	1
2017	0	0	FALL	F	0	F	0	fail	0	fail	0
2020	3	1	SPRING	B+	3.3	B	3	pass	1	pass	1
2020	3	1	SPRING	D	1	D	1	fail	0	fail	0

1.3 Summary of Previous Work

The results from data collected from the Math Placement Test offered on a volunteer basis to incoming students in the fall of 2017 were combined with information available from student records and analyzed by Dr. Dwight Lauer of Silvics Analytics in Wingate, NC. The recommendations for using the Math Placement Test to improve the placement of students into math courses was summarized in the table below and is printed in the Academic Course Catalog 2019 to be used as a reference by students and faculty advisors when deciding on a student course of study.

Fig 1.3.1

Prerequisite Course Recommendations

Course	Criteria Group 1	Criteria Group 2	Criteria Group 3
Math 112		Math SAT < 500 **AND** Placement < 20	ACT < 20 **AND** Placement < 23
Math 115	Math SAT ≥ 500 **OR** ACT ≥ 20 **OR** Math 112	Math SAT < 500 **AND** 20 ≤ Placement < 23	ACT < 20 **AND** 23 ≤ Placement < 26
Math 117 **Math 120**	Placement ≥ 26 **OR** Math 115	Math SAT < 500 **AND** Placement ≥ 23	ACT < 20 **AND** Placement ≥ 26
Math 209	Placement ≥ 20		

	OR Math 112 **OR** Math 115 **OR** Math 116 **OR** Math 117 **OR** Math 120		

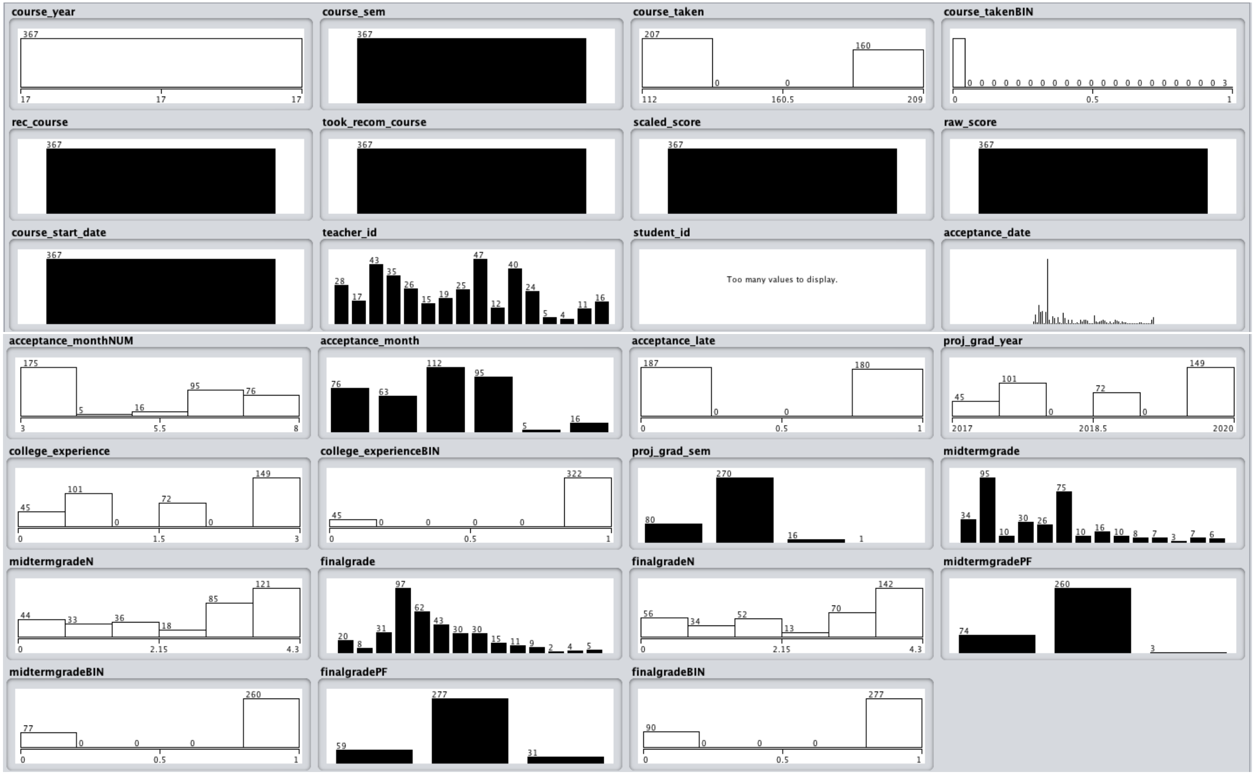
2.1 Data Mining Techniques Applied

After unifying the datasets, the author wanted to look at the distributions of each of the attributes for a record. The 2017 dataset was split into two sets, one containing information for students that did not take the Math Placement test and one containing the information for the students that did take the Math Placement test. In a way, this was the authors attempt at creating two clusters of students, starting will the whole set and using one attribute to branch into two clusters and then stopping.

Weka vs. 3.8.4 was used to load the .csv files and to "visualize all":

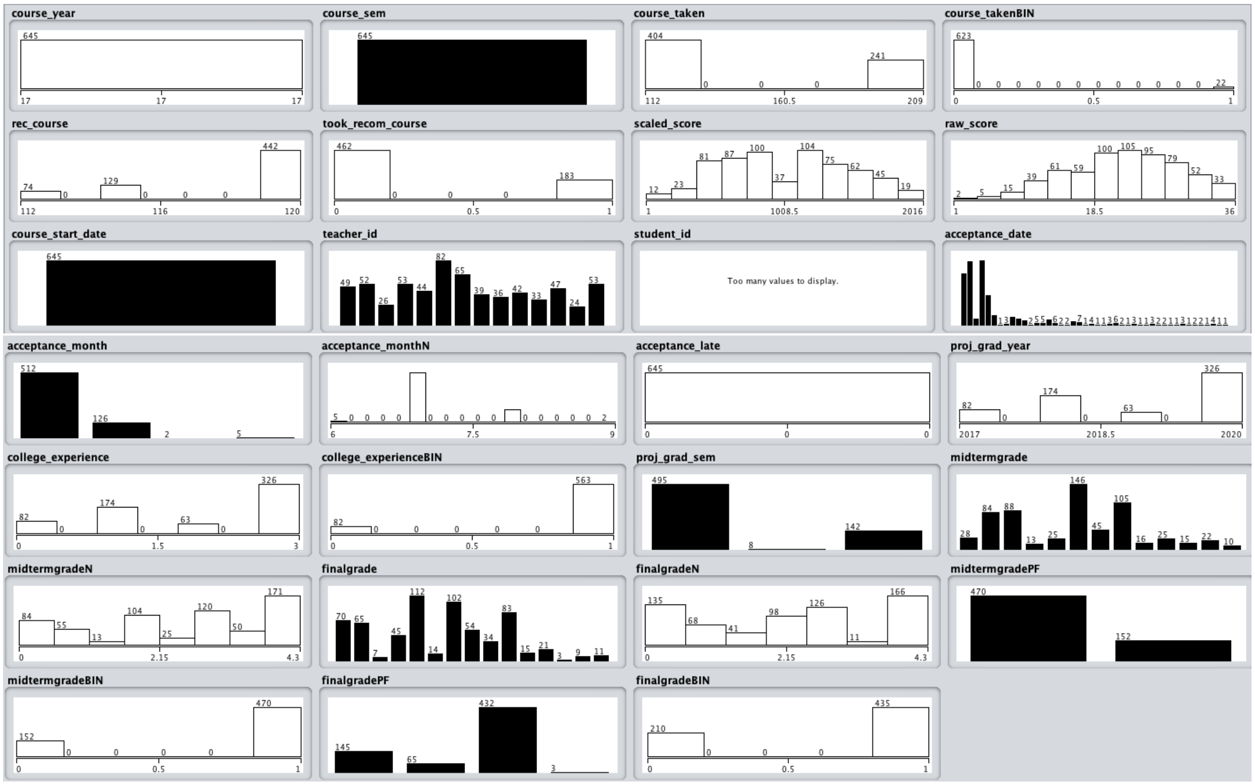
Visualize all for NoMathPlacement2017extended.csv attributes:

Fig. 2.1.1



Visualize all for MathPlacement2017extended.csv attributes:

Fig. 2.1.2

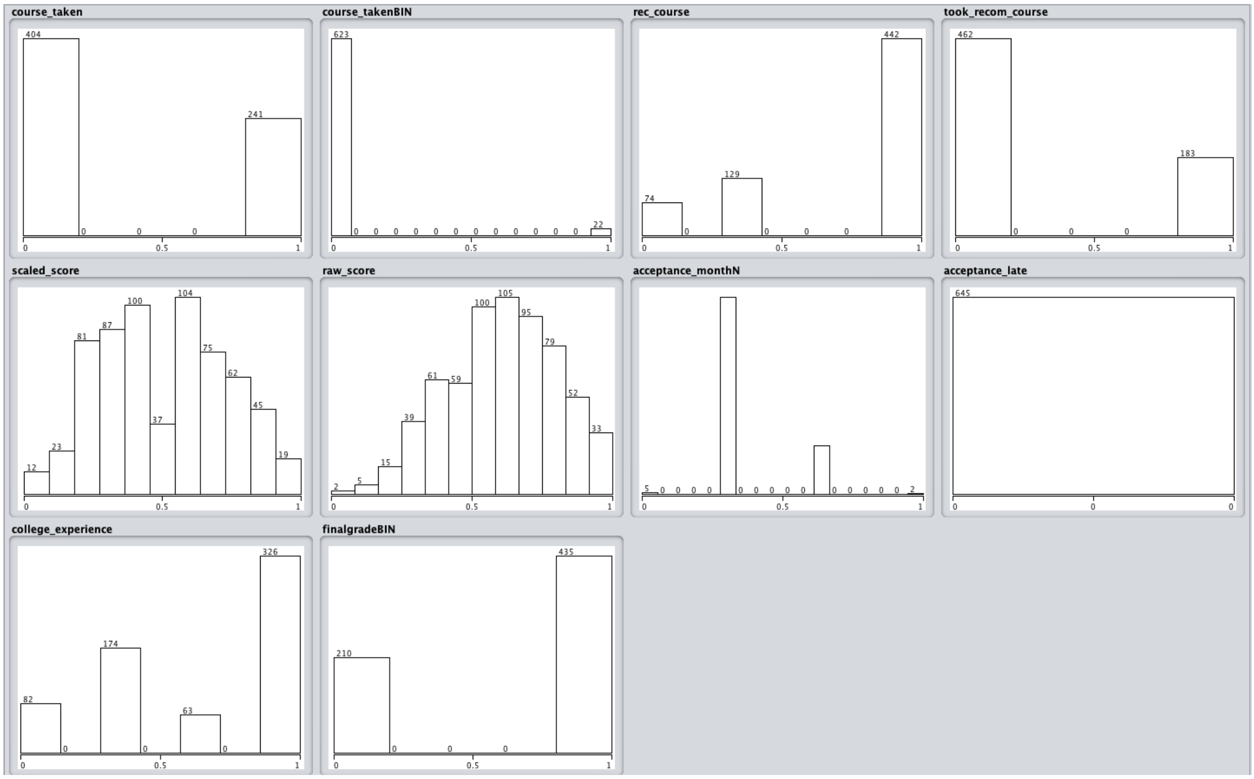


Too many values to display.

One observation made looking at these data sets is that of the students taking the Math Placement Test, very few of these students (in fact none) were accepted after May 2017. Of the students who did not take the Math Placement Test, roughly half were accepted after May 2017. For the majority of students entering into Wingate straight from high school, the college application process begins in September of their senior year. Wingate University has a "rolling application date" policy, meaning that there is no fixed submission date. If students are applying to other schools with deadlines for applications, it is more likely for the students to complete these applications first. In addition, students who have not yet achieved their "goal" standardized test score(s) will take these standardized tests later in the senior year to attempt their highest score after completion of high school courses. It stands to reason that more strong candidates for college courses are among the early acceptance subclass of these 2017 students than in the "accepted_late" category.

2017 Placement Taken and choosing selected numeric attributes, applying the normalizing filter in Weka:

Fig. 2.1.3



For fun, I ran the simple linear regression (Classify) on this data with the predicted y-hat value the pass or fail ranging from 0 to 1. As anticipated the correlation coefficient r^2 was rather low, indicating that the variation in the pass/fail of students is not well accounted for with a single variable. In particular, as a linear function of the variable "raw_score" on the Math Placement Test, the correlation coefficient r^2 is 0.2884, meaning that only 28% of the variation in whether a student is "more likely" to pass or fail (independent of any other variable) is accounted for in a linear relationship between the raw score and pass/fail variable. The "highest" not-really linear association came from the regression on pass/fail versus a student's college experience with a correlation coefficient r^2 of 0.3471. The majority of the incoming 2017 students who took the Math Placement Test had some college experience either from taking an AP Class with an acceptable grade on their AP test or through transfer credits earned from another college or university. It would seem logical that students who have had some experience with the level of expectations and pace of a college level course would be more likely to pass a college level course than students with no college experience.

Figure 2.1.4

```
=== Run information ===

Scheme:       weka.classifiers.functions.SimpleLinearRegression
Relation:     MathPlacement2017extended-weka.filters.unsupervised.attribute.Remove-R1-2,9-
13,16,18-20,22,24,26-weka.filters.unsupervised.attribute.Remove-R10-12-
weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0-
weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0-
weka.filters.unsupervised.attribute.Remove-R1-5,7-9
Instances:    645
Attributes:   2
              raw_score
              finalgradeBIN
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Linear regression on raw_score

0.69 * raw_score + 0.25

Predicting 0 if attribute value is missing.


Time taken to build model: 0 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.2884
Mean absolute error                      0.4019
Root mean squared error                  0.4487
Relative absolute error                 91.3585 %
Root relative squared error             95.5905 %
Total Number of Instances              645
```

Figure 2.1.5

```
=== Run information ===

Scheme:       weka.classifiers.functions.SimpleLinearRegression
Relation:     MathPlacement2017extended-weka.filters.unsupervised.attribute.Remove-R1-2,9-
13,16,18-20,22,24,26-weka.filters.unsupervised.attribute.Remove-R10-12-
weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0-
weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0
Instances:    645
Attributes:   10
              course_taken
              course_takenBIN
              rec_course
              took_recom_course
              scaled_score
              raw_score
              acceptance_monthN
              acceptance_late
              college_experience
              finalgradeBIN
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Linear regression on college_experience

0.45 * college_experience + 0.38

Predicting 0 if attribute value is missing.


Time taken to build model: 0.01 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                0.3471
Mean absolute error                    0.3841
Root mean squared error                  0.4395
Relative absolute error               87.3129 %
Root relative squared error             93.6345 %
Total Number of Instances               645
```

Again, for experimentation, I used 10 numerical attributes and the RegressionByDiscretization filter under "meta" subdirectory of Classify. Not knowing what really to expect, I was pleased to see that a pruned J48 tree had been created that appears to have height of nine, 43 nodes and 22 leaves and ran with a quick build time of 0.06 seconds. The attributes had already been normalized, and this process discretized the values of each attribute into what seems to be at most 10 bins…although I am not sure if they are created using equal number, equal width or some other criterial. The coefficients for the regression model are not given, but the correlation coefficient is 0.3534 with a relative absolute error of approximately 80%. So, as far as using numerical attributes to produce a regression model, things did not work out well for the 2017 data from students who took the Math Placement Test. I will mention that I attempted running regressions using SPSS, but normalizing data is done more by first using the analytics to find the mean and standard deviation for each attribute then entering in those values reported back into a second computation. Thus, Weka made this process much more direct. It would have been nice to see a scatter plot of the data and the regression lines on the same graph, but maybe another day especially since linear regressions do not seem to be the best predictive "classifier."

Figure 2.1.6

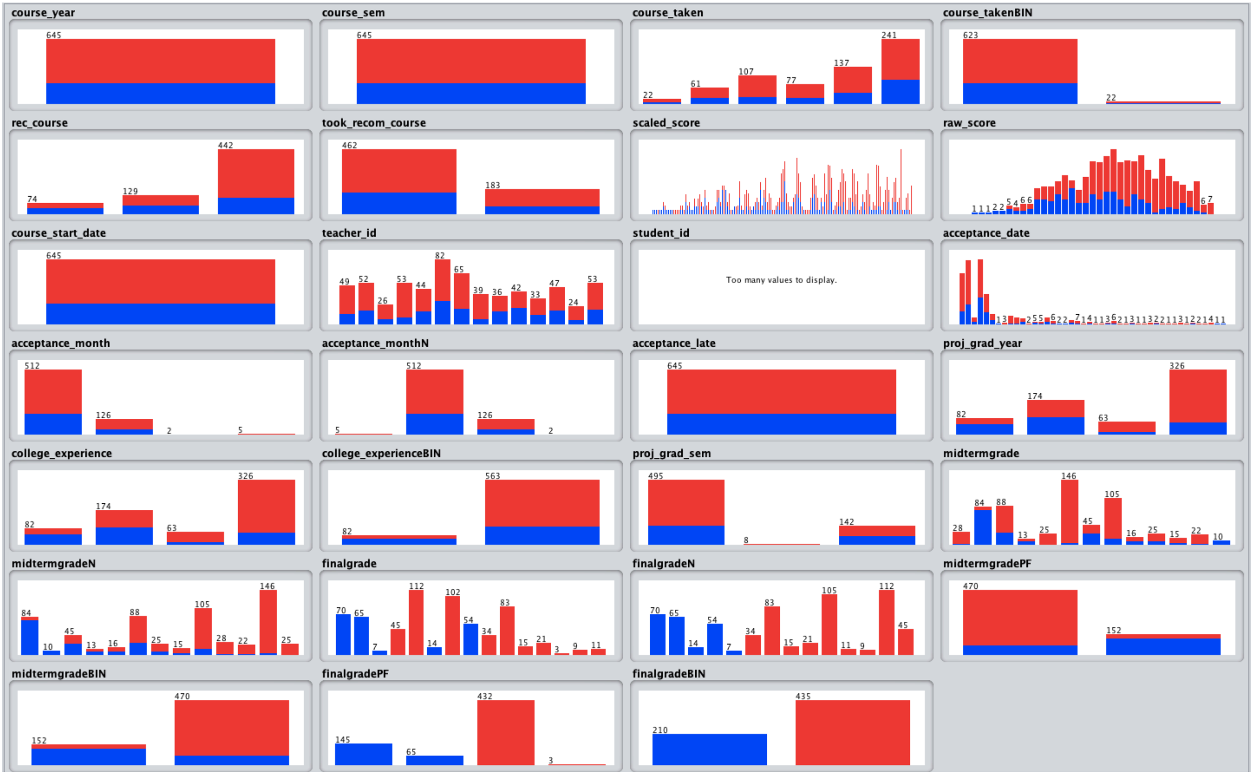
```
=== Run information ===

Scheme:       weka.classifiers.meta.RegressionByDiscretization -B 10 -K
weka.estimators.UnivariateEqualFrequencyHistogramEstimator -W weka.classifiers.trees.J48 --
-C 0.25 -M 2
Relation:     MathPlacement2017extended-weka.filters.unsupervised.attribute.Remove-R1-2,9-
13,16,18-20,22,24,26-weka.filters.unsupervised.attribute.Remove-R10-12-
weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0-
weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0
Instances:    645
Attributes:   10
              course_taken
              course_takenBIN
              rec_course
              took_recom_course
              scaled_score
              raw_score
              acceptance_monthN
              acceptance_late
              college_experience
              finalgradeBIN
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Regression by discretization

Class attribute discretized into 10 values

Classifier spec: weka.classifiers.trees.J48 -C 0.25 -M 2
J48 pruned tree
------------------

college_experience <= 0.333333
|   rec_course <= 0.375
|   |   course_takenBIN <= 0
|   |   |   course_taken <= 0.041237
|   |   |   |   course_taken <= 0.030928
|   |   |   |   |   scaled_score <= 0.202978: '(0.9-inf)' (2.0)
|   |   |   |   |   scaled_score > 0.202978: '(-inf-0.1]' (11.0/2.0)
|   |   |   |   course_taken > 0.030928
|   |   |   |   |   scaled_score <= 0.304715
|   |   |   |   |   |   rec_course <= 0
|   |   |   |   |   |   |   college_experience <= 0: '(-inf-0.1]' (2.0)
|   |   |   |   |   |   |   college_experience > 0
|   |   |   |   |   |   |   |   raw_score <= 0.142857: '(-inf-0.1]' (2.0)
|   |   |   |   |   |   |   |   raw_score > 0.142857: '(0.9-inf)' (7.0/1.0)
|   |   |   |   |   |   rec_course > 0: '(0.9-inf)' (8.0/1.0)
|   |   |   |   |   scaled_score > 0.304715: '(-inf-0.1]' (4.0)
|   |   |   course_taken > 0.041237: '(-inf-0.1]' (59.0/14.0)
|   |   course_takenBIN > 0
|   |   |   scaled_score <= 0.204963: '(-inf-0.1]' (4.0)
|   |   |   scaled_score > 0.204963: '(0.9-inf)' (8.0/1.0)
|   rec_course > 0.375
|   |   college_experience <= 0
|   |   |   course_taken <= 0.030928: '(0.9-inf)' (4.0/1.0)
|   |   |   course_taken > 0.030928: '(-inf-0.1]' (48.0/20.0)
|   |   college_experience > 0
|   |   |   took_recom_course <= 0
|   |   |   |   raw_score <= 0.8
|   |   |   |   |   acceptance_monthN <= 0.333333
|   |   |   |   |   |   raw_score <= 0.457143: '(-inf-0.1]' (2.0)
|   |   |   |   |   |   raw_score > 0.457143: '(0.9-inf)' (50.0/13.0)
|   |   |   |   |   acceptance_monthN > 0.333333
|   |   |   |   |   |   raw_score <= 0.685714
|   |   |   |   |   |   |   scaled_score <= 0.401489: '(0.9-inf)' (2.0)
|   |   |   |   |   |   |   scaled_score > 0.401489: '(-inf-0.1]' (4.0)
|   |   |   |   |   |   raw_score > 0.685714: '(0.9-inf)' (5.0)
|   |   |   |   raw_score > 0.8
|   |   |   |   |   raw_score <= 0.942857: '(-inf-0.1]' (9.0/1.0)
|   |   |   |   |   raw_score > 0.942857: '(0.9-inf)' (2.0)
|   |   |   took_recom_course > 0
|   |   |   |   raw_score <= 0.914286: '(-inf-0.1]' (20.0/7.0)
|   |   |   |   raw_score > 0.914286: '(0.9-inf)' (3.0)
college_experience > 0.333333: '(0.9-inf)' (389.0/72.0)

Number of Leaves  :   22

Size of the tree :    43


Time taken to build model: 0.06 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.3534
Mean absolute error                      0.3545
Root mean squared error                  0.4477
Relative absolute error                 80.6034 %
Root relative squared error             95.3788 %
Total Number of Instances              645
```

Changing directions and applying the NumericToNomial filter on all the attributes, it was interesting to re-evaluate the conditional distribution of pass (red)/ fail (blue) given each of the other attributes.   I must say, that I was more pleased with color versus the black and white viewport produced from "visualize all" earlier. It would be nice to figure out how to make the target class value of "pass" blue and the value of "fail" red, but I adjusted.

On first glance, the bar graph of the variable "finalgrade" seemed to be inconsistent with expected given that the x-axis values are from {W, I, F, D-, D+, D, C-, …, A, A+}. When hovering the cursor over the bars, it was discovered that the terms are not in lexicographical order. It made much more visual sense to look at the values in the modal for the value "finalgradeN" (transformed with the filter from numeric to nominal) and are in order along the x-axis {0.1, 0.1, 0, 0.7, 1.0, 1.3, 1.7, 2, 2.3, 2.7,…,4, 4.3}. The distribution for "midtermgradeN" and "rawscore" both are as expected, with these variables being positively correlated with pass/fail target class. This meaning that the higher the midterm grade or raw score, the more likely the student is to pass the class they took. It would be interesting to look at the subpopulation of those students who took the course recommended or less versus the subpopulation of students who elected to disregard the recommendation in 2017 and took a course "higher." Looking at the distribution of pass (red) in the bar graph for "took_recom_course" suggest that there could be a different distribution of "pass" among those who did take the recommended course given the percentage of pass in the column on the left is visually larger than the percentage of red (pass) in the column on the right.

Other items of notice would be the bars for each teacher id. It appears that the majority of instructors of these classes have the same pass/fail percentage for their classes, with a couple of teachers with high pass rates. It is unclear whether these instuctors' pass rates are influenced by the level of class that they are teaching (say MATH 120, MATH117 = Calculus, Calculus for Business Majors) or if this rate is perhaps the result of class size or teaching experience level.

Fig 2.1.7



The filter for creating a Bayes Network using a 10-fold cross validation for the training set/test set, the results below use ALL of the attributes available for students who had taken the Math Placement Test. As expected, the classifier scores well. Starting with the Confusion Matrix, most of the records are correctly judged with most of the instances recorded along the main diagonal. This means that the number of true positives is high (precision – sensitivity) of 0.995. The F-score and ROC-area are both “high” values: 0.998 and 1 respectively.

Figure 2.1.8

```
=== Run information ===

Scheme:       weka.classifiers.bayes.BayesNet -D -Q weka.classifiers.bayes.net.search.local.K2 --
-P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5
Relation:     MathPlacement2017extended-
weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last
Instances:    645
Attributes:   27
              course_year
              course_sem
              course_taken
              course_takenBIN
              rec_course
              took_recom_course
              scaled_score
              raw_score
              course_start_date
              teacher_id
              student_id
              acceptance_date
              acceptance_month
              acceptance_monthN
              acceptance_late
              proj_grad_year
              college_experience
              college_experienceBIN
              proj_grad_sem
              midtermgrade
              midtermgradeN
              finalgrade
              finalgradeN
              midtermgradePF
              midtermgradeBIN
              finalgradePF
              finalgradeBIN
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Bayes Network Classifier
not using ADTree
#attributes=27 #classindex=26
Network structure (nodes followed by parents)
course_year(1): finalgradeBIN
course_sem(1): finalgradeBIN
course_taken(6): finalgradeBIN
course_takenBIN(2): finalgradeBIN
rec_course(3): finalgradeBIN
took_recom_course(2): finalgradeBIN
scaled_score(150): finalgradeBIN
raw_score(35): finalgradeBIN
course_start_date(1): finalgradeBIN
teacher_id(14): finalgradeBIN
student_id(612): finalgradeBIN
acceptance_date(44): finalgradeBIN
acceptance_month(4): finalgradeBIN
acceptance_monthN(4): finalgradeBIN
acceptance_late(1): finalgradeBIN
proj_grad_year(4): finalgradeBIN
college_experience(4): finalgradeBIN
college_experienceBIN(2): finalgradeBIN
proj_grad_sem(3): finalgradeBIN
```

```
midtermgrade(13): finalgradeBIN
midtermgradeN(13): finalgradeBIN
finalgrade(15): finalgradeBIN
finalgradeN(14): finalgradeBIN
midtermgradePF(2): finalgradeBIN
midtermgradeBIN(2): finalgradeBIN
finalgradePF(4): finalgradeBIN
finalgradeBIN(2):
LogScore Bayes: -24237.417143225975
LogScore BDeu: -38438.916367726946
LogScore MDL: -34352.9862821778
LogScore ENTROPY: -28333.34886239932
LogScore AIC: -30194.34886239932


Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         644               99.845  %
Incorrectly Classified Instances         1                0.155  %
Kappa statistic                          0.9965
Mean absolute error                      0.0029
Root mean squared error                  0.0315
Relative absolute error                  0.6601 %
Root relative squared error              6.7227 %
Total Number of Instances              645

=== Detailed Accuracy By Class ===

             TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
Class
             1.000    0.002    0.995      1.000    0.998      0.996    1.000     1.000     0
             0.998    0.000    1.000      0.998    0.999      0.996    1.000     1.000     1
Weighted Avg.    0.998    0.001    0.998      0.998    0.998      0.996    1.000     1.000

=== Confusion Matrix ===

   a   b   <-- classified as
 210   0 |   a = 0
   1 434 |   b = 1
```

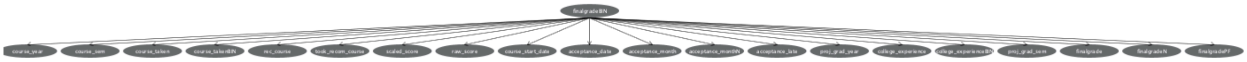
In order to have a useful classifier to use when a student has not yet taken the course, I remove the variables related to midterm grades and teacher_id. When I reran the classifier BayesNetwork, the model is too good as in it correctly classifies 100%.???

Fig. 2.1.9

```
=== Run information ===

Scheme:       weka.classifiers.bayes.BayesNet -D -Q weka.classifiers.bayes.net.search.local.K2 --
-P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5
Relation:     MathPlacement2017extended-
weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-
weka.filters.unsupervised.attribute.Remove-R10-11,20-21,24-25
Instances:    645
Attributes:   21
              course_year
              course_sem
              course_taken
              course_takenBIN
              rec_course
              took_recom_course
              scaled_score
              raw_score
              course_start_date
              acceptance_date
              acceptance_month
              acceptance_monthN
              acceptance_late
              proj_grad_year
              college_experience
              college_experienceBIN
              proj_grad_sem
              finalgrade
              finalgradeN
              finalgradePF
              finalgradeBIN
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Bayes Network Classifier
not using ADTree
#attributes=21 #classindex=20
Network structure (nodes followed by parents)
course_year(1): finalgradeBIN
course_sem(1): finalgradeBIN
course_taken(6): finalgradeBIN
course_takenBIN(2): finalgradeBIN
rec_course(3): finalgradeBIN
took_recom_course(2): finalgradeBIN
```

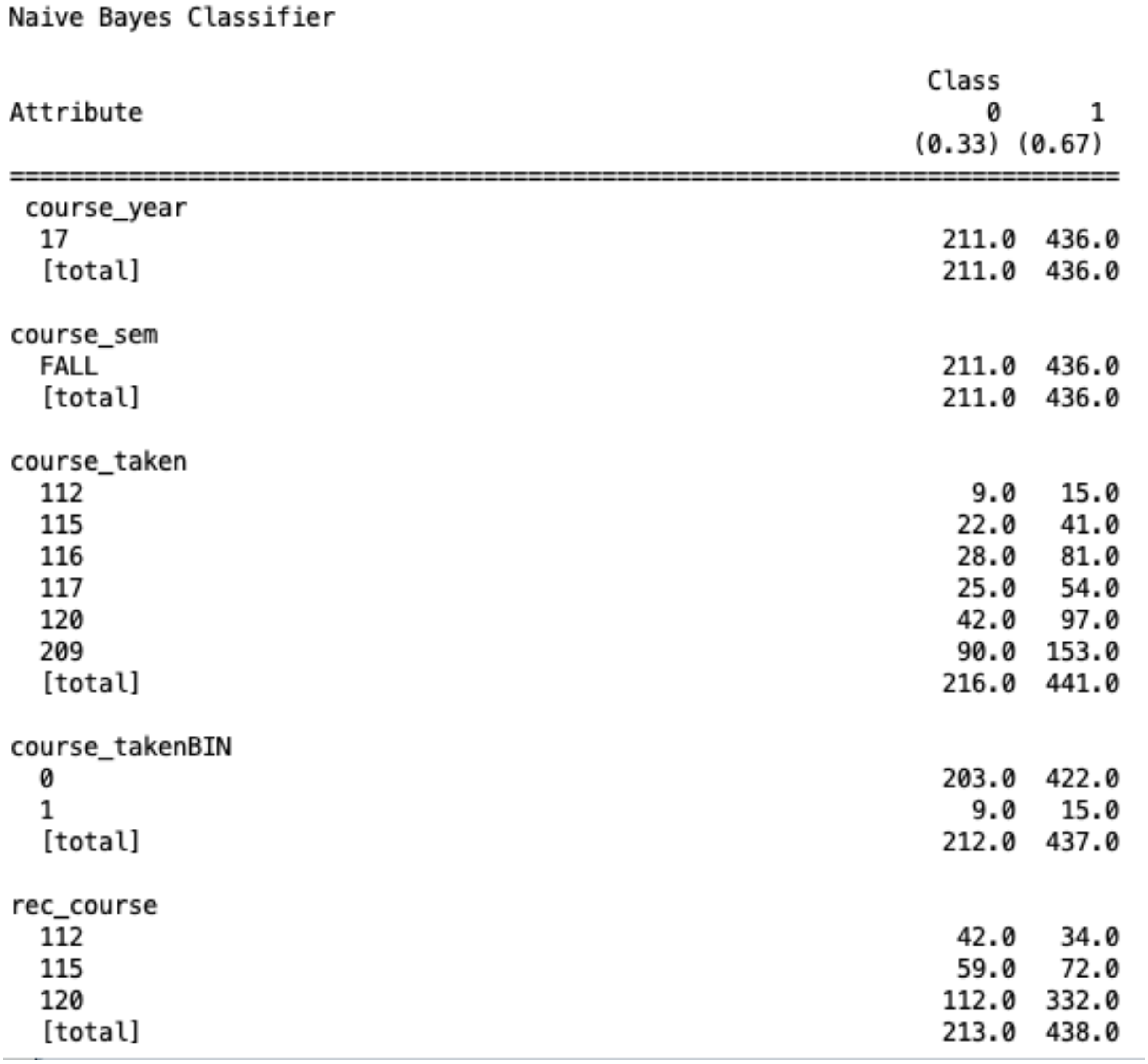
```
scaled_score(150): finalgradeBIN
raw_score(35): finalgradeBIN
course_start_date(1): finalgradeBIN
acceptance_date(44): finalgradeBIN
acceptance_month(4): finalgradeBIN
acceptance_monthN(4): finalgradeBIN
acceptance_late(1): finalgradeBIN
proj_grad_year(4): finalgradeBIN
college_experience(4): finalgradeBIN
college_experienceBIN(2): finalgradeBIN
proj_grad_sem(3): finalgradeBIN
finalgrade(15): finalgradeBIN
finalgradeN(14): finalgradeBIN
finalgradePF(4): finalgradeBIN
finalgradeBIN(2):
LogScore Bayes: -14947.256004933815
LogScore BDeu: -18010.50656001243
LogScore MDL: -17373.78637747328
LogScore ENTROPY: -15559.161663612058
LogScore AIC: -16120.161663612058


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         645               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0.0006
Relative absolute error                  0.009  %
Root relative squared error              0.1206 %
Total Number of Instances              645

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
Class
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     0
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     1
Weighted Avg.    1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000

=== Confusion Matrix ===
   a   b   <-- classified as
 210   0 |   a = 0
   0 435 |   b = 1
```

And then, I look at the graph and notice that it is rather "short." It turns out that the previous model is essentially the same tree. My level of understanding of what to do at this point is lacking, so I will table this for further study.

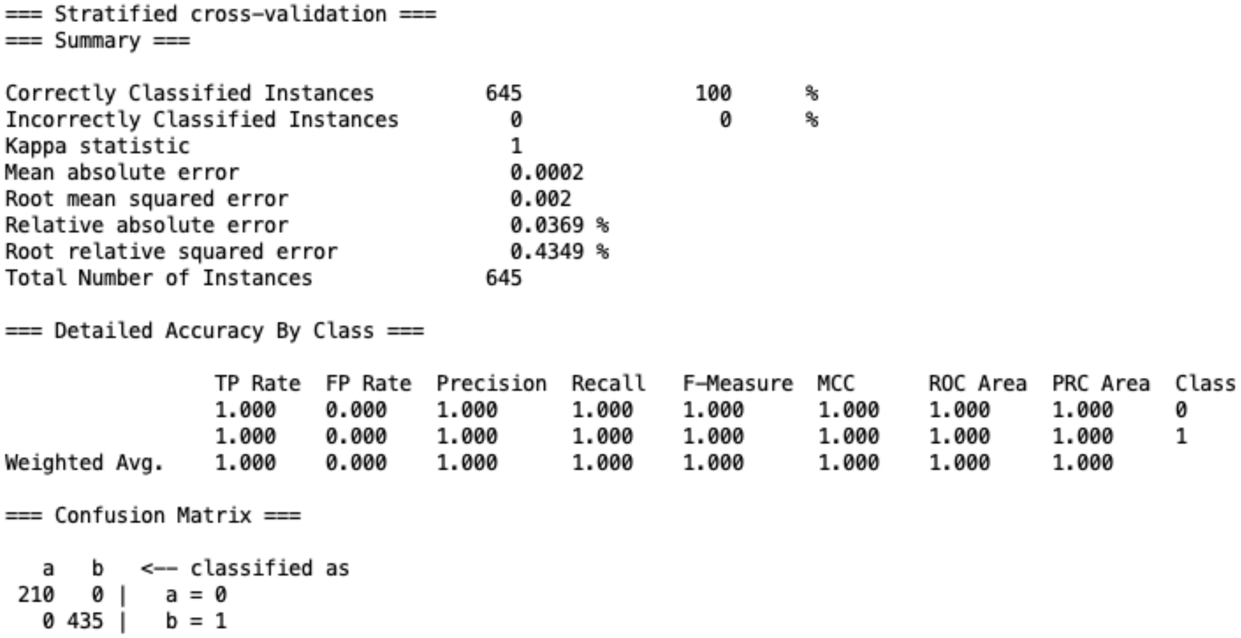


The Naïve Bayes Classifier produces a "deeper" tree and the output is much too long to include here, so I have reproduced a portion of the top of the tree in "list" format, the summary of accuracy and confusion matrix. Which all seem to indicate the model is "good." Again, I will put the evaluation of the model as a subject of further study.

Fig. 2.1.10



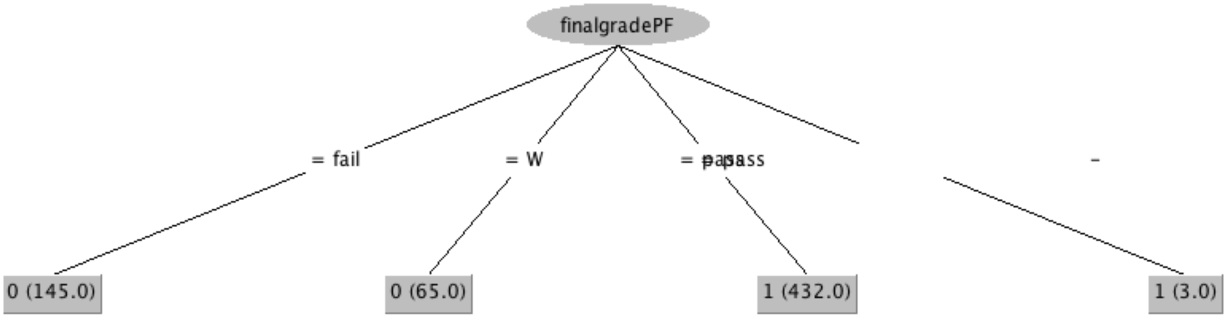
```
Naive Bayes Classifier

                                                                  Class
Attribute                                                             0      1
                                                                 (0.33) (0.67)
===============================================================================
 course_year
  17                                                              211.0  436.0
  [total]                                                         211.0  436.0

course_sem
  FALL                                                            211.0  436.0
  [total]                                                         211.0  436.0

course_taken
  112                                                               9.0   15.0
  115                                                              22.0   41.0
  116                                                              28.0   81.0
  117                                                              25.0   54.0
  120                                                              42.0   97.0
  209                                                              90.0  153.0
  [total]                                                         216.0  441.0

course_takenBIN
  0                                                               203.0  422.0
  1                                                                 9.0   15.0
  [total]                                                         212.0  437.0

rec_course
  112                                                              42.0   34.0
  115                                                              59.0   72.0
  120                                                             112.0  332.0
  [total]                                                         213.0  438.0
```



```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         645              100      %
Incorrectly Classified Instances         0                0      %
Kappa statistic                          1
Mean absolute error                      0.0002
Root mean squared error                  0.002
Relative absolute error                  0.0369 %
Root relative squared error              0.4349 %
Total Number of Instances              645

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     0
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     1
Weighted Avg.    1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000

=== Confusion Matrix ===

   a   b   <-- classified as
 210   0 |   a = 0
   0 435 |   b = 1
```

The J48 Tree classifier filter on the same 10 nominal variables produces the following pruned tree with 5 nodes including 4 leaves (so not deep at all, height of one):

Fig. 2.1.11



When I switched to clustering..forming clusters of students who are similar to each other, I first chose the K-mean method allowing Weka to randomly seed the clusters. I did wonder if it was possible to maybe start with more than 2 clusters, and would offer that this would make another direction for future study.

Fig. 2.1.12

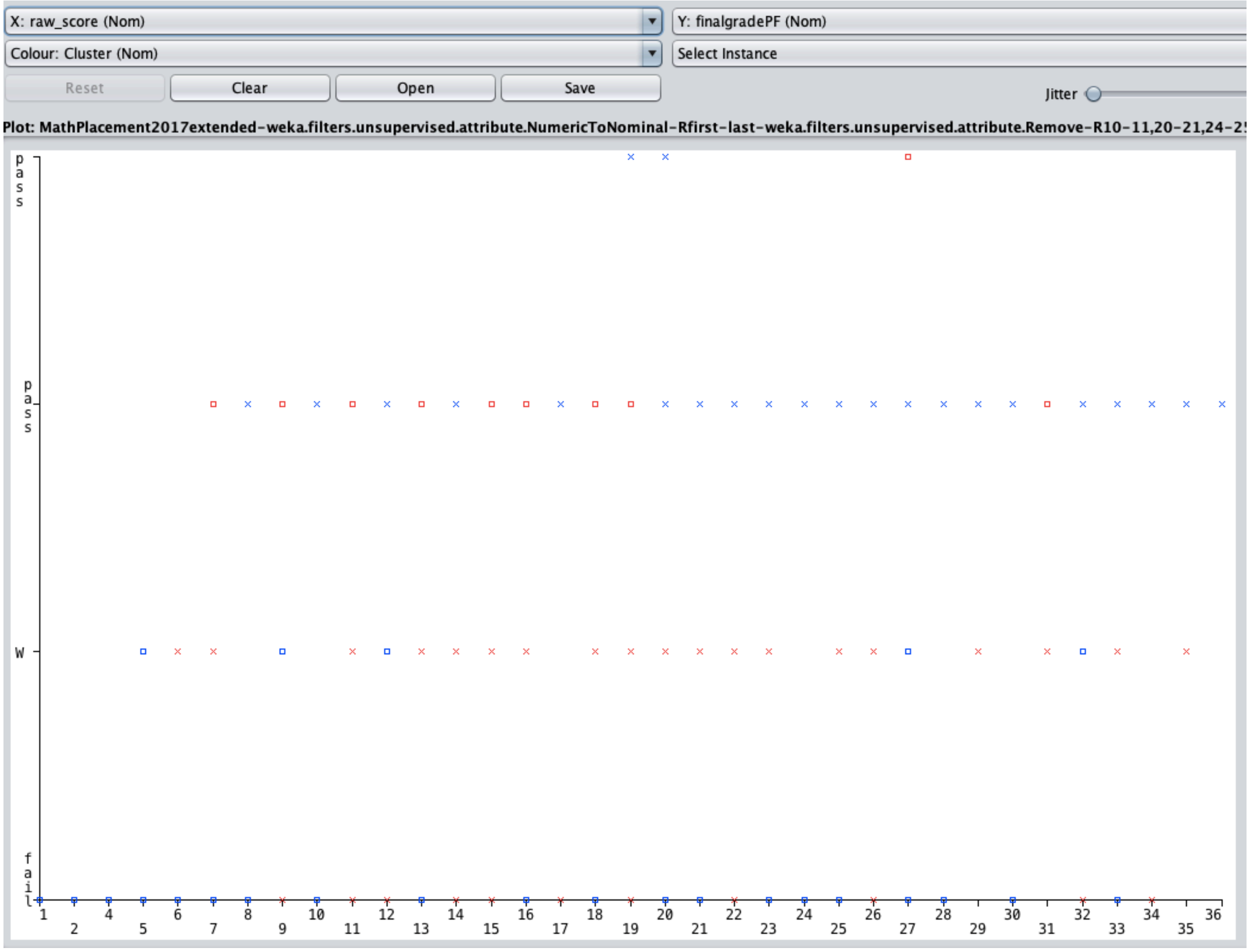
```
=== Clustering model (full training set) ===


kMeans
======

Number of iterations: 4
Within cluster sum of squared errors: 4841.0

Initial starting points (random):

Cluster 0: 17,FALL,116,0,112,0,203,5,21-Aug-17,14-Jul-
17,Jul,7,0,2018,1,1,SUMMER,F,0,fail,0
Cluster 1: 17,FALL,112,1,112,1,106,7,21-Aug-17,17-Jul-
17,Jul,7,0,2018,1,1,SPRING,W,0.1,W,0

Missing values globally replaced with mean/mode

Final cluster centroids:
                                       Cluster#
Attribute                             Full Data         0         1
                                        (645.0)   (477.0)   (168.0)
======================================================================
====================
course_year                                 17        17        17
course_sem                                FALL      FALL      FALL
course_taken                               209       209       120
course_takenBIN                              0         0         0
rec_course                                 120       120       120
took_recom_course                            0         0         1
scaled_score                              1816       912      1816
raw_score                                   22        21        26
course_start_date                    21-Aug-17 21-Aug-17 21-Aug-17
acceptance_date                      17-Jul-17 14-Jul-17 17-Jul-17
acceptance_month                           Jul       Jul       Jul
acceptance_monthN                            7         7         7
acceptance_late                              0         0         0
proj_grad_year                            2020      2020      2020
college_experience                           3         3         3
college_experienceBIN                        1         1         1
proj_grad_sem                           SPRING    SPRING    SPRING
finalgrade                                   A         A         W
finalgradeN                                  4         4       0.1
finalgradePF                              pass      pass      pass
finalgradeBIN                                1         1         1
```

```
Time taken to build model (full training data) : 0.04 seconds

=== Model and evaluation on test split ===

kMeans
======

Number of iterations: 5
Within cluster sum of squared errors: 3114.0

Initial starting points (random):

Cluster 0: 17,FALL,209,0,120,0,911,20,21-Aug-17,17-Jul-17,Jul,7,0,2018,1,1,SPRING,C-
,1.7,pass,1
Cluster 1: 17,FALL,120,0,115,0,610,16,21-Aug-17,14-Jul-
17,Jul,7,0,2020,3,1,SPRING,C,2,pass,1

Missing values globally replaced with mean/mode

Final cluster centroids:
                                        Cluster#
Attribute                           Full Data        0          1
                                     (425.0)    (229.0)    (196.0)
==========================================================================
====================
course_year                              17         17         17
course_sem                             FALL       FALL       FALL
course_taken                            209        209        120
course_takenBIN                           0          0          0
rec_course                              120        120        120
took_recom_course                         0          0          0
scaled_score                           1214        812       1111
raw_score                                22         20         22
course_start_date                 21-Aug-17  21-Aug-17  21-Aug-17
acceptance_date                   17-Jul-17  17-Jul-17  14-Jul-17
acceptance_month                        Jul        Jul        Jul
acceptance_monthN                         7          7          7
acceptance_late                           0          0          0
proj_grad_year                         2020       2018       2020
college_experience                        3          1          3
college_experienceBIN                     1          1          1
proj_grad_sem                        SPRING     SPRING     SPRING
```

```
finalgrade                                   A       A       B
finalgradeN                                  4       4       3
finalgradePF                               pass    pass    pass
finalgradeBIN                                1       1       1




Time taken to build model (percentage split) : 0.01 seconds

Clustered Instances

0      121 ( 55%)
1       99 ( 45%)
```

The first time I ran the K-means cluster algorithm no evaluation statistics were produced. Before I ran a second time, I continued a little further with this model. The two clusters seem to be roughly equal in size, and that made me wonder if any patterns could be visualized in the clusters if the raw_score of the Math Placement Test was graphed against the target class of fingalgradePF which would be a pass or a fail for a student. The visualization is as follows and offers yet another opportunity for future research.

Fig. 2.1.13



X: raw_score (Nom)

Y: finalgradePF (Nom)

Colour: Cluster (Nom)

Plot: MathPlacement2017extended-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.unsupervised.attribute.Remove-R10-11,20-21,24-2

Knowing that the seeds will determine the new clusters, I do run the K-means filter again and produce a result of 2 clusters with the following data that shows that 36.2% of the test instances were incorrectly classified.

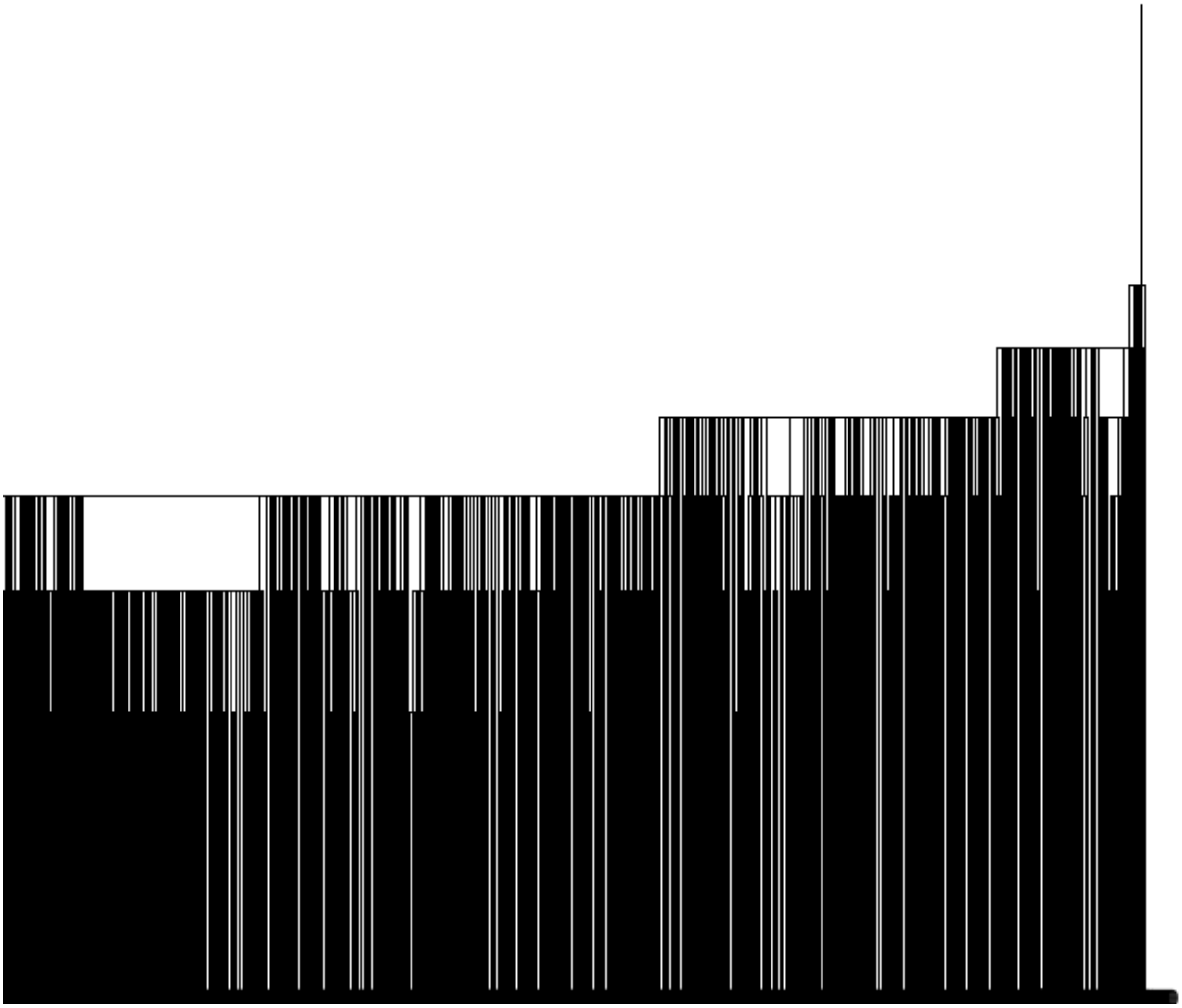
Fig. 2.1.14

```
Time taken to build model (full training data) : 0.01 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      477 ( 74%)
1      168 ( 26%)


Class attribute: finalgradeBIN
Classes to Clusters:

  0   1  <-- assigned to cluster
```

```
 138  72 | 0
 339  96 | 1

Cluster 0 <-- 1
Cluster 1 <-- 0

Incorrectly clustered instances :         234.0    36.2791 %
```

And one more heuristic for classifying the records: A Hierarchical Cluster. Building up from the many individual instances to one we have a dendogram that is not as cluttered as some that I have seen…

Fig. 2.1.15



Section 3 Conclusions and Future Directions

3.1 Conclusions

I have learned that there is a lot of work at the start of a project. In order to access the data, I needed to write a proposal for the Research Review Board at Wingate University and fulfil their requirement for Social, Behavior and Educational Research Involving Human Subjects. The data is often in different formats when you do receive it with little explanation of the variables. It was interesting to learn how SPSS and Weka accept data and to "play" with the features of each of these software packages. I definitely prefer Weka and truly enjoyed the courses for Weka available at https://www.cs.waikato.ac.nz/ml/weka/courses.html led by Professor Witten. It is also clear to me now that data mining requires patients, a willingness to research and learn from others, and more patients.

Section 3.2. Directions for Future Study.

The data that I was able to normalize and organize for analysis consisted of a third of the data that is on my computer. It is definitely needed to continue this work to evaluate the patterns that may exist in each piece of the data and in the data collectively. At my current introductory level of understanding of the topic in Data Mining, I could not adequately create models and assess their performance, efficiency, sensitivity, specificity, and other measures that are extremely important when working with classification problems.

Bibliography: