COMP306 Group 4
April 30, 2021

Student Test Performance

Emilia Lambert, Sophia Sklar, Corinne Steuk

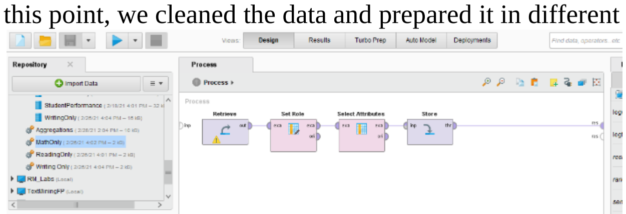
Introduction

Our main goal is to analyze different conclusions about influences of student performance on a particular exam that tests three subject areas: math, reading, and writing. We are posing issues and questions in regard to how a student's demographic information may influence his or her performance on a test. Further, we wonder how accurately we can predict different attributes given the test scores, or can we predict a test score given a set of attributes. We will also analysis through data visualizations multiple correlations between test scores and student attributes in a variety of combinations. There are 8 distinct attributes. The attributes include *gender* (Female or Male), *race* (Group A, B, C, D, E), *parental level of education* (range from some high school to master's degree), *type of lunch* (free reduced or standard), *test preparation* (completed or not completed), and *test scores* (for math, reading and writing with a range of numerical scores between 0 and 100). The scores seem to be the best labels to classify on. It could be useful to discretize these scores into categories (such as high, medium, and low) to use them in the classification methods. To use some methods that require only numerical values, we will have to use the Nominal to Numerical process to change these attributes to numerical values. For example, test preparation changes to completed:1, and none:0. We are also included an aggregate score attribute that would act like a total score over all three exams.

We chose three different classification methods to test our data: Naïve Bayes, K-NN, and Decision Trees. Using different types allowed us to give us our project investigation multiple dimensions. In each phase, from Data Exploration to Initial Process to Classifications, we were able to draw conclusions about both current trends and future predictive models using this data set.

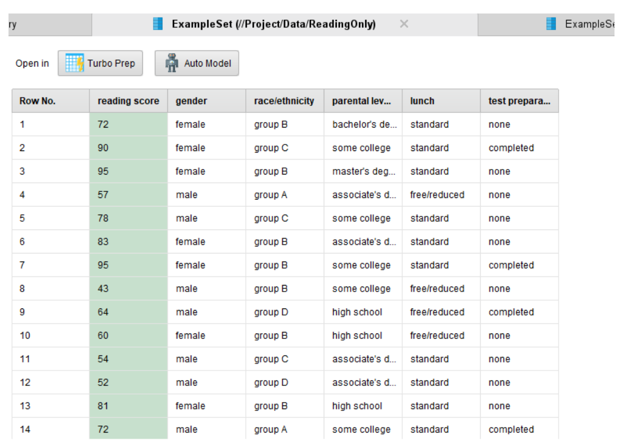
Data Exploration & Initial Processes

We began our project with simple processes to filter, aggregate, and transform the data for initial data exploration. Prior to applying algorithms and model, we thoroughly investigated the data to learn more about the dataset as a whole. We performed some preliminary classifications at the end of this phase, but more importantly at this point, we cleaned the data and prepared it in different ways to be able to perform the methods later.



This process filters out two of the subject attributes, so we can look at how the polynominal attributes effect one subject at a time. In this picture, we set the label attribute role to Math and removed the Reading and Writing attribute. We have two other processes that do the same for the other two subjects.

The picture to the right shows the output of the above process. This dataset, however, is the result of the Reading Only process (the above shows the Math Only process).

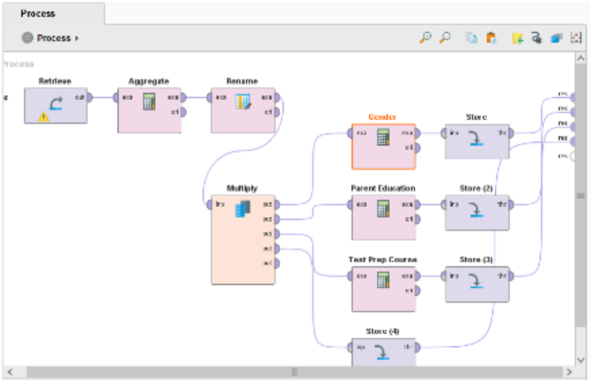


ExampleSet (//Project/Data/ReadingOnly)

Row No.	reading score	gender	race/ethnicity	parental lev...	lunch	test prepara...
1	72	female	group B	bachelor's de...	standard	none
2	90	female	group C	some college	standard	completed
3	95	female	group B	master's deg...	standard	none
4	57	male	group A	associate's d...	free/reduced	none
5	78	male	group C	some college	standard	none
6	83	female	group B	associate's d...	standard	none
7	95	female	group B	some college	standard	completed
8	43	male	group B	some college	free/reduced	none
9	64	male	group D	high school	free/reduced	completed
10	60	female	group B	high school	free/reduced	none
11	54	male	group C	associate's d...	standard	none
12	52	male	group D	associate's d...	standard	none
13	81	female	group B	high school	standard	none
14	72	male	group A	some college	standard	completed

ExampleSet (1,000 examples, 1 special attribute, 5 regular attributes)

Next, we made an Aggregate process to group and average all the scores by the different attributes. We store the results in our repository so we can retrieve them and use them later in our project. It is also interesting to see the overall averages by gender, parent education level, and test prep course. This is the process where we multiply the aggregated data to average it different ways.



The output of the below left is the full aggregated data by just average the three subjects (no group by). Then, on the right, we take this dataset and aggregate it again, but group by only the parent education level. We repeated this for both gender and test prep course as well.

As expected, the average grades for the Math, Reading, and Writing scores are all highest with a higher parent level of education. Our group was surprised, however, that the scores were not overall higher, at least for the parent level of education in the college range. Even with a parent who has a master's degree, the averages for each score are only about 70%, 75%, and 76%, respectively. We expected a higher average in all aggregations overall; yet, we do not know the age of students or context of the tests to confirm our assumption.

Row No.	Gender	parental lev...	test prepara...	Math	Reading	Writing
1	female	associate's d...	completed	70.048	79.714	81.738
2	female	associate's d...	none	62.527	70.946	69.608
3	female	bachelor's de...	completed	71	80.682	83
4	female	bachelor's de...	none	66.927	75.463	75.902
5	female	high school	completed	61.897	71.241	72.379
6	female	high school	none	58.215	66.846	64.154
7	female	master's deg...	completed	69.857	81.286	82.786
8	female	master's deg...	none	64.364	73.955	74.364
9	female	some college	completed	67.929	78.262	79.500
10	female	some college	none	64.013	70.947	71.039
11	female	some high sc...	completed	63.829	74.943	75.486
12	female	some high sc...	none	56.464	65.464	63.786
13	male	associate's d...	completed	73.700	72.450	71.650
14	male	associate's d...	none	68.985	64.394	61.621

Row No.	parental lev...	average(Mat...	average(Re...	average(Wri...
1	associate's d...	68.815	71.876	71.154
2	bachelor's de...	70.043	73.366	73.953
3	high school	62.961	65.695	64.182
4	master's deg...	70.565	75.058	75.790
5	some college	68.359	70.873	70.484
6	some high sc...	64.021	67.512	65.608

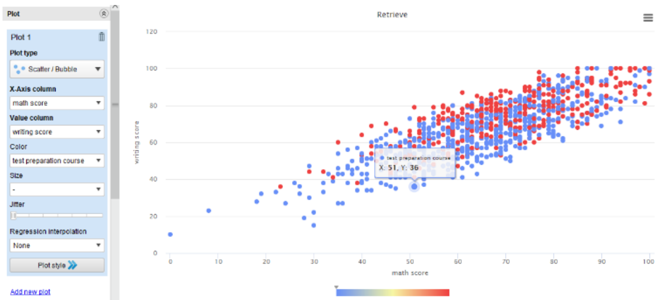
The next process we created was a correlation matrix to see the relationship between different attributes. We have the label attribute set to math score, but we could easily change this to reading or writing with a "Set Role" process. For this process to work, we had to change all the attribute types from nominal to numerical. We did this by a "Generate Attributes" and "Nominal to Numerical" processes. For simplicity, we changed the parent education level to numerical values in Excel with the lowest level, 'some high school,' set to 1 and the highest level, 'master's degree,' set to 6.

We note from the results that the reading and writing attributes have the highest correlation, which makes sense since these subjects are similar so students' performance on the test would be closely related. It is also interesting to point out that test preparation is not closely correlated with parental education level or lunch. We can conclude that student's financial stability or home situation do not closely influence their willingness to participate in the test prep course.

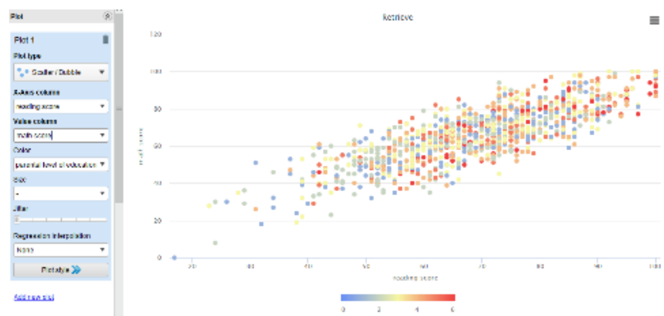
Row No.	math score	test prepara...	parental lev...	reading score	writing score	lunch
1	72	0	5	72	74	1
2	69	1	3	90	88	1
3	90	0	6	95	93	1
4	47	0	4	57	44	0
5	76	0	3	78	75	1
6	71	0	4	83	78	1
7	88	1	3	95	92	1
8	40	0	3	43	39	0
9	64	1	2	64	67	0
10	38	0	2	60	50	0
11	58	0	4	54	52	1
12	40	0	4	52	43	1
13	65	0	2	81	73	1
14	78	1	3	72	70	1

Attribut...	test pre...	parenta...	reading ...	writing ...	lunch
test prep...	1	-0.007	0.242	0.313	-0.017
parental ...	-0.007	1	0.191	0.237	-0.023
reading ...	0.242	0.191	1	0.955	0.230
writing s...	0.313	0.237	0.955	1	0.246
lunch	-0.017	-0.023	0.230	0.246	1

Since we now have numerical values for the attributes, we utilized scatter plots to evaluate trends. The first two visualizations we created use both math and reading for the axis values to give a sense of both subjects when comparing them with the other non-subject attributes (rather than using both reading and writing). The first evaluation was comparing the effect of completed the test preparation course.

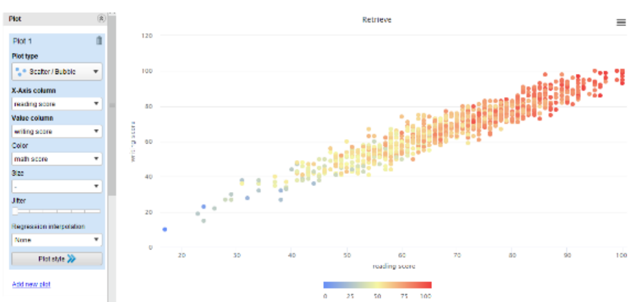


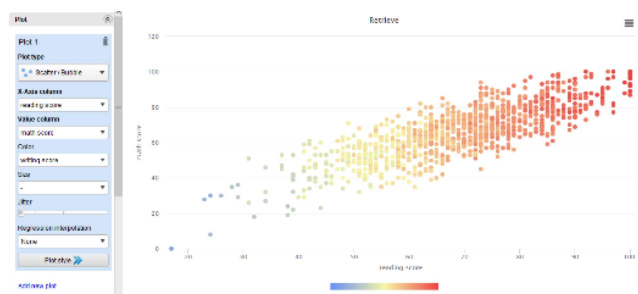
The x-axis is the math score, and the y-axis is the reading score. If the point is red, the student completed the test prep course, and a blue point means they did not. Although there is a slight increase in the number of red dots as x and y increase, it seems that this test prep course did not give the students as much help on the math course as it did the reading course. The majority of the red points fall around the 80 score for reading, but only the 65-70 score for math. While this is passing, most students who put the time into this course must have expected a better result.
The next graph was comparing the reading and math scores to parent level of education.



The math score is along the y axis and the reading score is along the x-axis. The more red the point is, the higher level of education a given student's parent has achieved. The coloration is more sporadic that we would have expected, but this shows that the parents' education does not play a large role in students' performance.

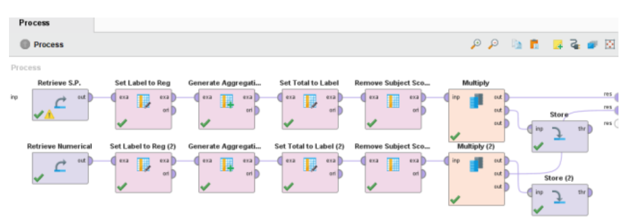
Next, we can look at all three subjects mapped against each other. The graph on the left shows reading and writing on the axes and math as the coloration. The graph on the right shows reading and math on the axis with writing on the coloration.





Looking at the spread of the data, we can see there is less variance with reading and writing (left) than there is with reading and math (right). Also, the left graph has a greater positive slope than the right. In conclusion, we can note that a student's reading and writing scores will be very similar. If they do well on the reading, they will probably do similarly well on the writing. This is not as much the case when comparing math and reading. This confirms the process and data received by the correlation matrix above.

We may consider using classification techniques on a total score, rather than the individual subject scores. So, we created a process that changes and saves a copy of the data set with a new column titled, 'Total Score,' with an average of the math, reading, and writing scores.



Before doing the aggregation, we changed the label attribute (which was the math score) to a regular attribute to perform this aggregation. Then, we set the 'Total Score' to be the new label attribute, so we can use it for predictions in the future. There are two rows of this process because we copied the same process for the numerical form of the dataset as well. From before, we created a data set with the test preparation values as 1 or 0 (completed/not), lunch as 1 or 0 (standard/reduced), and parental level of education 1 to 6 (some high school to master's degree). The attributes we left out of this data set were gender and ethnicity. Since we may need a numerical dataset with the total score for some classification methods, we included this in my Total Score Conversion process.

Row No.	Total Score	gender	race/ethnicity	parental lev...	lunch	test prepara...
1	72.667	female	group B	bachelor's de...	standard	none
2	82.333	female	group C	some college	standard	completed
3	92.667	female	group B	master's deg...	standard	none
4	49.333	male	group A	associate's d...	free/reduced	none
5	76.333	male	group C	some college	standard	none
6	77.333	female	group B	associate's d...	standard	none
7	91.667	female	group B	some college	standard	completed
8	40.667	male	group B	some college	free/reduced	none
9	65	male	group D	high school	free/reduced	completed
10	49.333	female	group B	high school	free/reduced	none
11	54.667	male	group C	associate's d...	standard	none
12	45	male	group D	associate's d...	standard	none
13	73	female	group B	high school	standard	none
14	73.333	male	group A	some college	standard	completed
15	53.667	female	group A	master's deg...	standard	none

Row No.	Total Score	test prepara...	parental lev...	lunch
1	72.667	0	5	1
2	82.333	1	3	1
3	92.667	0	6	1
4	49.333	0	4	0
5	76.333	0	3	1
6	77.333	0	4	1
7	91.667	1	3	1
8	40.667	0	3	0
9	65	1	2	0
10	49.333	0	2	0
11	54.667	0	4	1
12	45	0	4	1
13	73	0	2	1

<u>Classifications</u>

1. Naïve Bayes

Naïve Bayes is a statistical classifier that performs probabilistic prediction and predicts class membership. Based on Bayes' Theorem, this type of supervised learning technique is one of the more simpler classification methods, but it helps in building the fast machine learning models that can make quick predictions. Also, this method assumes the occurrence of a certain feature is independent of the occurrence of other features which is called class-conditional independence (and hence the name 'naïve'). For these reasons, we have decided Naïve Bayes may be a good classification process to use for our project data set.

Naïve Bayes is an incremental method, meaning that each training example can incrementally increase or decrease the probability that a hypothesis is correct. So, this prior knowledge can be combined with the observed data. This method uses probabilities to classify; more specifically, the Bayes' Theorem. The classifier will predict that a given tuple belongs to a certain class having the highest posterior probability conditioned on that tuple. Thus, the probability of a class, conditioned on the given tuple, is maximized (maximum posteriori hypothesis). In general, the classification is to determine P(class(m)|tuple(i)) or the probability that the hypothesis holds given the observed data sample tuple.

accuracy: 54.80% +/- 5.01% (micro average: 54.80%)

	true female	true male	class precision
pred. female	305	239	56.07%
pred. male	213	243	53.29%
class recall	58.88%	50.41%	

We first ran a Naïve Bayes classification on a data set we created that has the sum of each score as attribute 'Total Score' and deleting the other three subject scores. We switched the label attribute to gender to answer the question of how we can determine a student's gender based on their background and scores. As the performance vector shows, this did not produce the best, most accurate results. We believe this was due to the fact that the Total Score has such a wide variety of values that this effected the probability calculations for this attribute. We then discretized the three subject scores and ran the method again.

class names	upper limit
fail	50.0
barely pass	70.0
pass	85.0
high	92.0
excellent	100.0

We discretized by user specification so we could set appropriate ranges for what we consider high and low scores. This was the first discretization we completed. Again, the label is set to gender and this now includes all three subjects, rather than a total score. We also removed the race/ethnicity attribute because we decided the other attributes are more interesting to compare and would have more of an effect on the questions we are considering.

accuracy: 66.90% +/- 6.26% (micro average: 66.90%)

	true female	true male	class precision
pred. female	337	150	69.20%
pred. male	181	332	64.72%
class recall	65.06%	68.88%	

Above is the performance vector from this process. Binning does seem to improve the accuracy of the model and it makes more sense in the context of the issue. Below is the sample output. We used cross-validation for all of the Naïve Bayes algorithms as well.

Row No.	gender	prediction(g...	confidence(...	confidence(f...	reading score	writing score	math score	parental lev...	lunch	test prepara...
1	male	male	0.602	0.398	pass	barely pass	pass	3	standard	completed
2	male	female	0.245	0.755	high	pass	excellent	3	standard	none
3	male	male	0.604	0.396	barely pass	barely pass	barely pass	4	standard	none
4	male	male	0.578	0.422	barely pass	barely pass	barely pass	4	free/reduced	none
5	female	female	0.410	0.590	barely pass	pass	barely pass	3	free/reduced	completed
6	male	male	0.615	0.385	barely pass	barely pass	barely pass	1	free/reduced	none
7	female	female	0.342	0.658	pass	high	pass	3	free/reduced	none
8	female	female	0.398	0.602	pass	pass	pass	4	standard	none
9	male	female	0.433	0.567	pass	pass	pass	2	standard	none
10	male	female	0.368	0.632	pass	pass	pass	4	free/reduced	completed
11	female	male	0.522	0.478	pass	barely pass	barely pass	3	standard	completed
12	female	male	0.601	0.399	barely pass	barely pass	barely pass	3	free/reduced	none
13	male	male	0.599	0.401	barely pass	barely pass	barely pass	4	standard	completed
14	male	male	0.753	0.247	barely pass	fail	barely pass	4	standard	none

While looking back at the binning techniques, we decided that our bins were too specific and possibly could be related to the still somewhat low accuracy of my model. We went back to create different, more general bins.

class names	upper limit
fail	50.0
pass	75.0
good	85.0
excellent	100.0

Row No.	gender	prediction(g...	confidence(...	confidence(f...	reading score	writing score	math score	parental lev...	lunch	test prepara...
1	male	male	0.691	0.309	pass	pass	good	3	standard	completed
2	male	female	0.244	0.756	excellent	good	excellent	3	standard	none
3	male	male	0.579	0.421	pass	pass	pass	4	standard	none
4	male	male	0.552	0.448	pass	pass	pass	4	free/reduced	none
5	female	male	0.571	0.429	pass	pass	pass	3	free/reduced	completed
6	male	male	0.590	0.410	pass	pass	pass	1	free/reduced	none
7	female	female	0.241	0.759	good	excellent	good	3	free/reduced	none
8	female	male	0.579	0.421	pass	pass	pass	4	standard	none
9	male	female	0.497	0.503	good	pass	pass	2	standard	none
10	male	female	0.306	0.694	good	good	good	4	free/reduced	completed
11	female	male	0.597	0.403	pass	pass	pass	3	standard	completed
12	female	male	0.576	0.424	pass	pass	pass	3	free/reduced	none
13	male	male	0.574	0.426	pass	pass	pass	4	standard	completed
14	male	male	0.748	0.252	pass	fail	pass	4	standard	none

Below is the new performance vector. While most percentages increased, including the overall accuracy, we were surprised to see the class recall for true female actually decrease by about 6%.

accuracy: 67.20% +/- 5.05% (micro average: 67.20%)

	true female	true male	class precision
pred. female	310	120	72.09%
pred. male	208	362	63.51%
class recall	59.85%	75.10%	

We decided to switch the discretization type from user specification to frequency with 5 bins. We thought possibly our previous parameters were not a good representation of the data.

Row No.	gender	prediction(g...	confidence(...	confidence(f...	reading score	writing score	math score	parental lev...
1	male	male	0.566	0.434	range3 [66.500 - 74.500]	range3 [65.500 - 73.500]	range4 [70.500 - 79.500]	3
2	male	female	0.200	0.800	range5 [82.500 - ∞]	range5 [81.500 - ∞]	range5 [79.500 - ∞]	3
3	male	male	0.721	0.279	range1 [-∞ - 57.500]	range2 [54.500 - 65.500]	range3 [62.500 - 70.500]	4
4	male	male	0.560	0.440	range2 [57.500 - 66.500]	range2 [54.500 - 65.500]	range2 [53.500 - 62.500]	4
5	female	male	0.521	0.479	range2 [57.500 - 66.500]	range3 [65.500 - 73.500]	range2 [53.500 - 62.500]	3
6	male	male	0.761	0.239	range1 [-∞ - 57.500]	range2 [54.500 - 65.500]	range2 [53.500 - 62.500]	1
7	female	female	0.130	0.870	range5 [82.500 - ∞]	range5 [81.500 - ∞]	range4 [70.500 - 79.500]	3
8	female	female	0.409	0.591	range3 [66.500 - 74.500]	range4 [73.500 - 81.500]	range4 [70.500 - 79.500]	4
9	male	male	0.516	0.484	range4 [74.500 - 82.500]	range3 [65.500 - 73.500]	range4 [70.500 - 79.500]	2
10	male	female	0.207	0.793	range4 [74.500 - 82.500]	range5 [81.500 - ∞]	range4 [70.500 - 79.500]	4
11	female	female	0.491	0.509	range3 [66.500 - 74.500]	range3 [65.500 - 73.500]	range3 [62.500 - 70.500]	3
12	female	male	0.568	0.432	range3 [66.500 - 74.500]	range2 [54.500 - 65.500]	range2 [53.500 - 62.500]	3
13	male	male	0.542	0.458	range2 [57.500 - 66.500]	range2 [54.500 - 65.500]	range3 [62.500 - 70.500]	4

This cross validation screenshot shows some of the ranges of each bin. The highest score bins have a minimum score of about 82, 81, and 79 for reading, writing, and math, respectively.

Below is the performance vector from this frequency-discretized Naïve Bayes algorithm. It has the highest accuracy so far and leads us to conclude the discretized by frequency is a worthy option.

accuracy: 68.40% +/- 5.44% (micro average: 68.40%)

	true female	true male	class precision
pred. female	368	166	68.91%
pred. male	150	316	67.81%
class recall	71.04%	65.56%	

Since the other attributes (parent education, lunch, and test prep course) were still factored into the model, we wondered if we could better predict gender if we just considered the three test scores. Below is the performance vector of this output. We continued to discretize by frequency with 5 bins as above. This model has the best accuracy so far.

accuracy: 69.70% +/- 5.01% (micro average: 69.70%)

	true female	true male	class precision
pred. female	374	159	70.17%
pred. male	144	323	69.16%
class recall	72.20%	67.01%	

Because a student cannot control their lunch/financial situation or their parents' level of income, we left these out. Since they can control whether they participate in the test preparation course, we decided to add this attribute back in to the model.

accuracy: 69.80% +/- 4.89% (micro average: 69.80%)

	true female	true male	class precision
pred. female	374	158	70.30%
pred. male	144	324	69.23%
class recall	72.20%	67.22%	

In comparison with the model above, this only had a slight increase in the accuracy of the model.

From our visualizations in initial exploration, we observed how males tend to perform better in math, while females perform better in reading and writing. From running these Naïve Bayes algorithms, we can conclude there is a possibility that a trend of test scores could be used to predict the students' gender. However, we would be cautious when using this prediction for any one student given there are many unique cases and these models still do not reach at least the 70% accuracy range.

2. K-NN

We used the k-NN classification to predict the different attributes from the dataset. First, for the integer attributes, we used a cross validation operation to predict the *math score*, *reading score*, *writing score*, and the *total test score*. The main process is shown in Figure 1. First, the student performance dataset that includes the *total score* attribute is retrieved. The set role operation is used to set an attribute to label. Each time the process is run, each test score attribute is set as the label. The subprocess for the cross validation operation is shown in Figure 2. This operation is split into two sections: training and testing. The k-NN classification is used for the training portion. K is set to 10 to optimize the accuracy of the predictions for this dataset. In the testing portion, the model is applied and the performance operation will calculate the squared and root mean squared error.

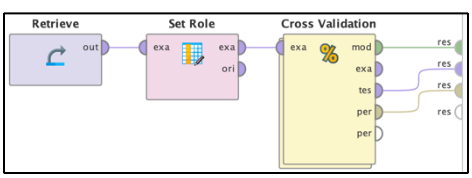


Figure 1: Cross Validation Process

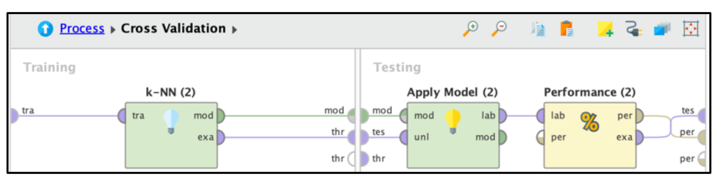


Figure 2: Cross Validation Subprocess

The prediction for each score and their errors are shown in Figures 3 through 6. The *total test score* prediction has the lowest root mean squared error and the writing score prediction has the lowest squared error. The *math score* prediction has the highest error for both calculations, so its predictions are the least accurate.

Row No.	math score	prediction(math score)
1	76	74.799
2	64	63.690
3	88	87.793
4	59	58.500
5	39	45.356
6	65	66.004
7	85	83.234
8	75	75.766
9	65	65.704
10	82	82.322
11	62	62.497
12	60	60.395
13	71	70.092
14	45	46.605
15	67	67.302

root_mean_squared_error: 2.589 +/- 0.531
squared_error: 6.957 +/- 3.263

Figure 3: Math Score Prediction

Row No.	reading score	prediction(reading score)
1	78	78.424
2	64	65.189
3	89	88.611
4	58	59.321
5	64	59.706
6	66	64.685
7	91	90.694
8	85	83.222
9	77	75.526
10	82	79.002
11	67	66.999
12	60	60.613
13	77	77.200
14	53	53.516
15	84	85.302

root_mean_squared_error: 2.384 +/- 0.240
squared_error: 5.737 +/- 1.216

Figure 4: Reading Score Prediction

Row No.	writing score	prediction(writing score)
1	75	74.387
2	67	66.607
3	86	85.496
4	59	57.881
5	57	58.089
6	62	62.878
7	89	90.362
8	82	81.995
9	74	74.114
10	74	77.583
11	69	68.090
12	60	58.401
13	77	76.297
14	55	53.102
15	86	83.305

root_mean_squared_error: 2.292 +/- 0.205
squared_error: 5.290 +/- 0.965

Figure 5: Writing Score Prediction

Row No.	totalTestScore	prediction(totalTestScore)
1	229	228.193
2	195	194.874
3	263	261.332
4	176	175.592
5	160	160.903
6	193	193.181
7	265	263.817
8	242	241.787
9	216	215.125
10	238	236.804
11	198	196.914
12	180	180.412
13	225	225.096
14	153	153.274
15	237	235.605

root_mean_squared_error: 2.082 +/- 1.415
squared_error: 6.137 +/- 10.375

Figure 6: Total Score Prediction

We also ran the performance operation on the polynominal attributes to predict their values. The overall process is shown in Figure 7. First the student performance dataset with the total test score attribute is retrieved. Next each attribute is set to a label. The k-NN classification is performed with k set to 10. The model is then applied and the performance operation outputs the attribute predictions and performance vectors.

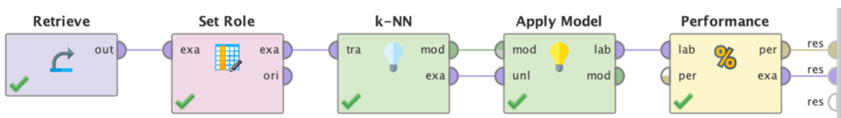


Figure 7: Performance Process

The performance vectors are shown for the attributes in Figures 8 through 12. The highest accuracy is calculated for the *gender* attribute at 90.5%. The lowest accuracy is calculated for the *parental level of education* attribute at 48.4%. The *gender* attribute has a higher accuracy because it only has two values (*male* or *female*) but *parental level of education* has six values (*some high school, high school, some college, associate's degree, bachelor's degree, master's degree*).

accuracy: 76.60%

	true none	true completed	class precision
pred. none	583	175	76.91%
pred. completed	59	183	75.62%
class recall	90.81%	51.12%	

Figure 8: Test Preparation Course Attribute Performance Vector

accuracy: 90.50%

	true female	true male	class precision
pred. female	476	53	89.98%
pred. male	42	429	91.08%
class recall	91.89%	89.00%	

Figure 9: Gender Attribute Performance Vector

accuracy: 77.60%

	true standard	true free/reduced	class precision
pred. standard	577	156	78.72%
pred. free/reduced	68	199	74.53%
class recall	89.46%	56.06%	

Figure 10: Lunch Attribute Performance Vector

accuracy: 48.40%

	true bachelor's deg...	true some college	true master's degree	true associate's de...	true high school	true some high sch...	class precision
pred. bachelor's d...	33	7	9	6	6	9	47.14%
pred. some college	32	143	14	42	28	40	47.83%
pred. master's deg...	3	1	9	3	2	1	47.37%
pred. associate's d...	22	37	17	130	28	29	49.43%
pred. high school	18	17	5	21	102	33	52.04%
pred. some high sc...	10	21	5	20	30	67	43.79%
class recall	27.97%	63.27%	15.25%	58.56%	52.04%	37.43%	

Figure 11: Parental Level of Education Attribute Performance Vector

accuracy: 52.30%

	true group B	true group C	true group A	true group D	true group E	class precision
pred. group B	89	21	17	20	18	53.94%
pred. group C	56	236	40	77	46	51.87%
pred. group A	2	3	8	3	1	47.06%
pred. group D	37	52	21	151	36	50.84%
pred. group E	6	7	3	11	39	59.09%
class recall	46.84%	73.98%	8.99%	57.63%	27.86%	

Figure 12: Race/Ethnicity Attribute Performance Vector

Gender has a higher accuracy because it is easier to guess one of two values than one of six. The *gender* attribute's value predictions and their confidences are shown in Figure 13. The *some college* value has the highest accuracy because it is the most common value of the attribute. Its value predictions and confidences are shown in Figure 14.

Row No.	gender	prediction(gender)	confidence(female)	confidence(male)
1	female	female	0.699	0.301
2	female	female	1	0
3	female	female	0.794	0.206
4	male	male	0.097	0.903
5	male	male	0.293	0.707
6	female	female	1	0
7	female	female	0.805	0.195
8	male	male	0.490	0.510
9	male	male	0.398	0.602
10	female	female	0.901	0.099
11	male	male	0.095	0.905
12	male	female	0.696	0.304
13	female	female	0.901	0.099
14	male	male	0.101	0.899
15	female	female	1	0
16	female	female	0.899	0.101

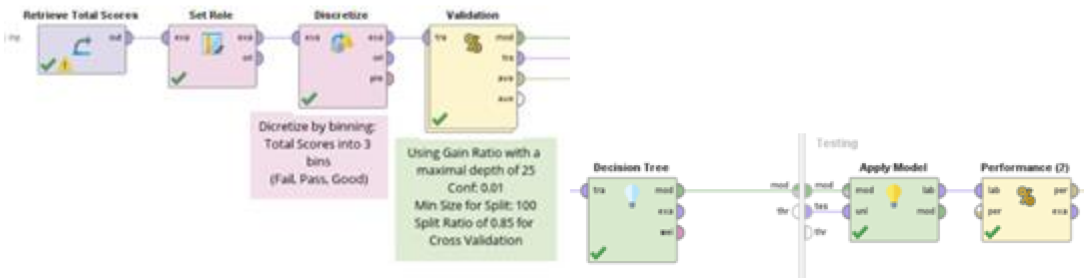
Figure 13: Gender Value Predictions

Row...	parental level of education	prediction(...	confidence(bachelor's...	confidence(some college)	confidence(master's...	confidence(associate'...	confidence(high school)	confidence(some high school)
1	bachelor's degree	some college	0.111	0.293	0.293	0.101	0.202	0
2	some college	some college	0.098	0.416	0	0.097	0.196	0.193
3	master's degree	some college	0.104	0.399	0.202	0.295	0	0
4	associate's degree	some high s...	0.099	0.102	0	0.208	0.193	0.398
5	some college	some college	0.100	0.307	0.097	0.200	0.097	0.199
6	associate's degree	some high s...	0	0.198	0	0.308	0.097	0.398
7	some college	some college	0.096	0.416	0.100	0.388	0	0
8	some college	some college	0.204	0.312	0	0.101	0.193	0.191
9	high school	some high s...	0.201	0.200	0.100	0	0.211	0.289
10	high school	associate's ...	0	0.096	0.197	0.302	0.212	0.194
11	associate's degree	high school	0.105	0.095	0	0.212	0.489	0.099
12	associate's degree	high school	0.096	0	0	0.209	0.498	0.197
13	high school	associate's ...	0	0.294	0	0.397	0.209	0.100
14	some college	some college	0	0.406	0.104	0.093	0.096	0.301
15	master's degree	associate's ...	0	0.101	0.111	0.396	0.296	0.096
16	some high school	some college	0.097	0.502	0	0.095	0.100	0.206

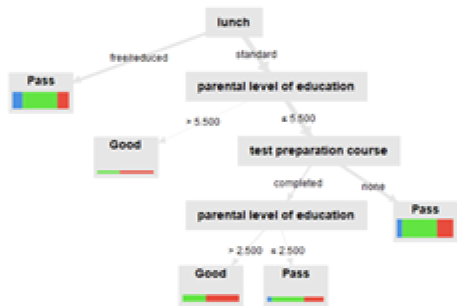
Figure 14: Parental Level of Education Values Prediction

The decision tree algorithm splits data into classes starting from a root node that can then be further split at later decision nodes. It classifies data using choices that are locally optimized, meaning that it makes decisions based on the information that it currently has available without considering previous or future information. It determines how to create branches using measures to determine which branch split will maximize the amount of information gained in that split of the data. Different techniques can be used to determine which attribute should have the most weight in determining the label class of the data. Pruning can be used to determine whether the training data has been too perfectly fit into different classes. This step is important because it makes sure that the test data will not be poorly classified by the model because the model was too specifically fit to the training data.

We created decision trees using the total score attribute as the label from the Total Score dataset. We discretized the values by binning the total scores into four ranges of equal size and then by user specification of bins (Fail <50, Pass <75, Good <100). For the splitting parameter of the decision tree, we tried gain ratio, information gain, GINI, and accuracy. We found that using Gain Ratio with a maximal depth of 25, a confidence of 0.01, and the minimum size for a split was 100. We used the cross-validation operator to measure the performance of the decision tree model. The process for this classification is shown below.



We are not super impressed with the accuracy of the decision tree. The two classes that it is predicting best are the score ranges 3 and 4. These ranges have the most values available to make predictions. The accuracy of the model is 61.33% which is not much better than random prediction. This low accuracy could be due to the inequality of samples in each of the bins of the total score categories since they were discretized by the user to represent specific score ranges instead of bins of equal size. The model does well at predicting the class with the most data which is the category Pass (51-75) as evidenced by the high recall of this class.



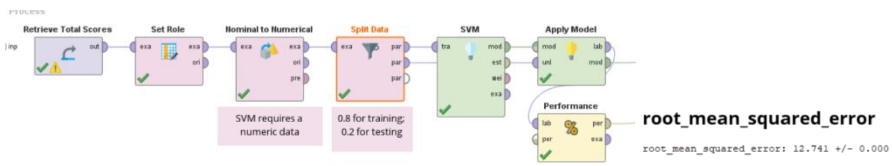
accuracy: 61.33%

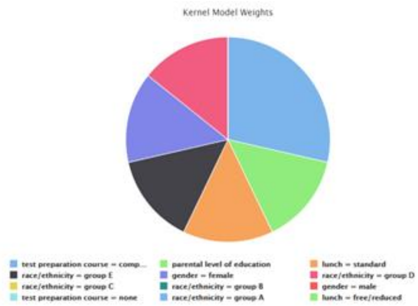
	true Fail	true Pass	true Good	class precision
pred. Fail	0	1	0	0.00%
pred. Pass	16	77	32	61.60%
pred. Good	0	9	15	62.50%
class recall	0.00%	88.51%	31.91%	

4. Support Vector Machine (SVM)

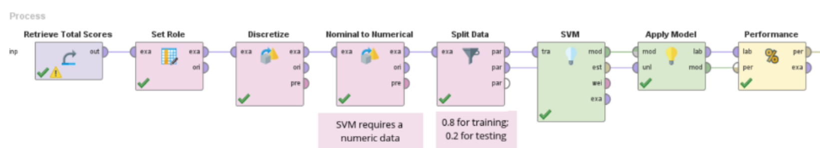
The Support Vector Machine algorithm classifies the data by looking for the best plane to separate the data into classes. It uses a technique called the maximum marginal hyperplane to find the boundaries of the classes that it is predicting. It can be used for both linear and non-linear classification models. The model looks for lines or hyperplanes that separate the data and attempts to create the largest margin between the classes. The data that is close to the lines (the data that has the largest influence on the margin) are called the support vectors. The algorithm uses weights to optimize the amount of space in the classes. The model then uses kernel separating function to transform the data into higher dimensions of data.

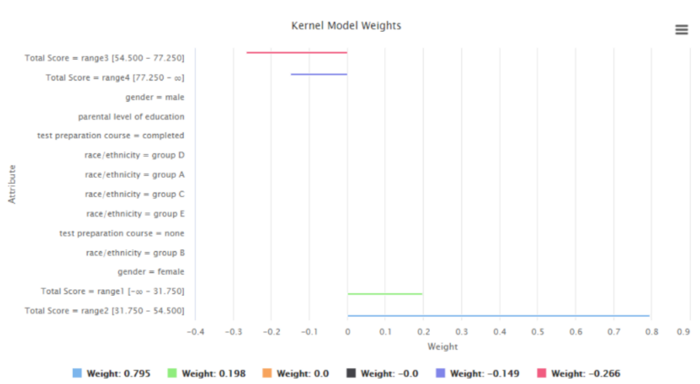
We used SVM to predict the total scores of the subjects using the dot kernel function. We converted the data from nominal to numeric as the model requires numeric data. We split the data for .8 to train and .2 to test the model. We used the optimize parameter operator in RapidMiner to find the best C value for the classification. We found that the C value of 0.036 worked the best for the classification. The model applied to the data shows a root mean squared error of 12.741. If the scores are binned as they were in the decision tree, this would mean that it is predicting the bins correctly. It is not very accurately predicting the total score though.





We then use the SVM to predict some of the polynomial attributes. The model that performed the best was the prediction of which lunch a student was receiving. For this model, the RapidMiner optimize tool changed the best C value of 1.688. The highest weight for this model was the Total Score attribute in Range 2 (31.750-54.500). This classification had an accuracy of 68.50%, which is better than random, but still not very reliable.





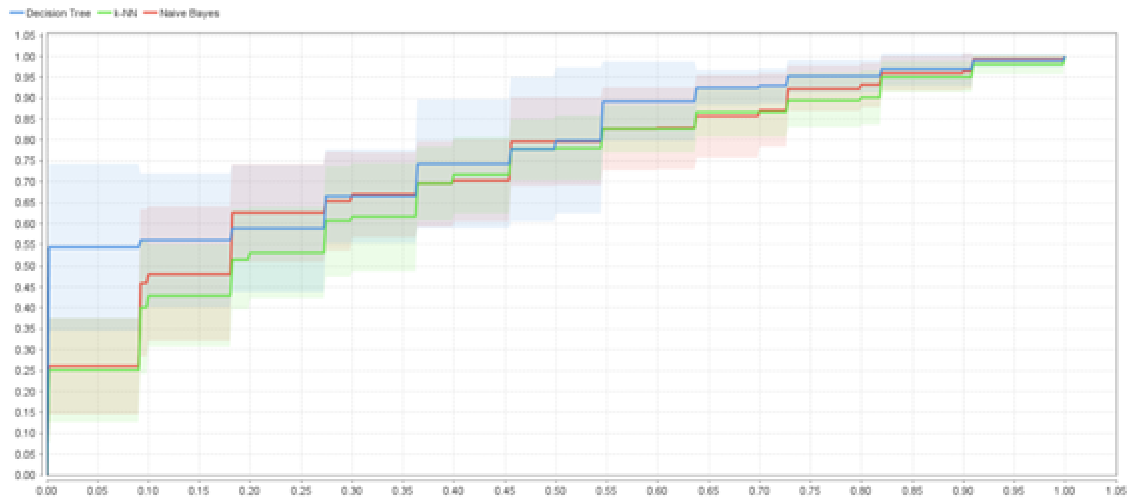
accuracy: 68.50%

	true standard	true free/reduced	class precision
pred. standard	115	49	70.12%
pred. free/reduced	14	22	61.11%
class recall	89.15%	30.99%	

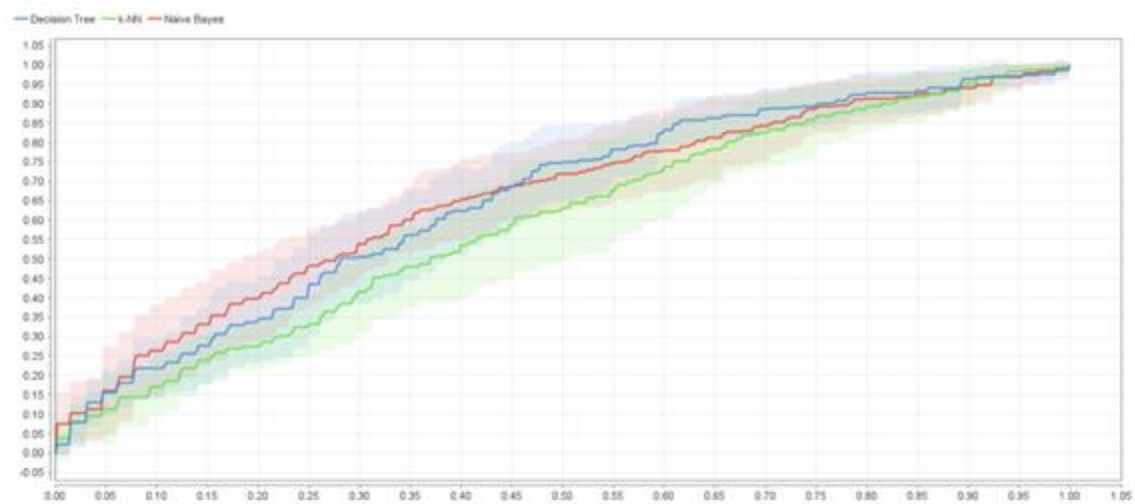
<u>Compare ROCs:</u>

We were also interested in comparing the performance of the classification methods that we chose against each other. To achieve this goal, we created ROC curves to compare the performance of the Naïve Bayes, k-NN, and decision tree model. We used the Total Score dataset to make these comparisons for the Total Score label attribute. For the gender as label attribute, we used the Total Score dataset with the three individual scores included. We wanted to explore which models perform the best when attempting to classify the data for two specific instances, specifically the Total Score attribute and the Gender attribute.

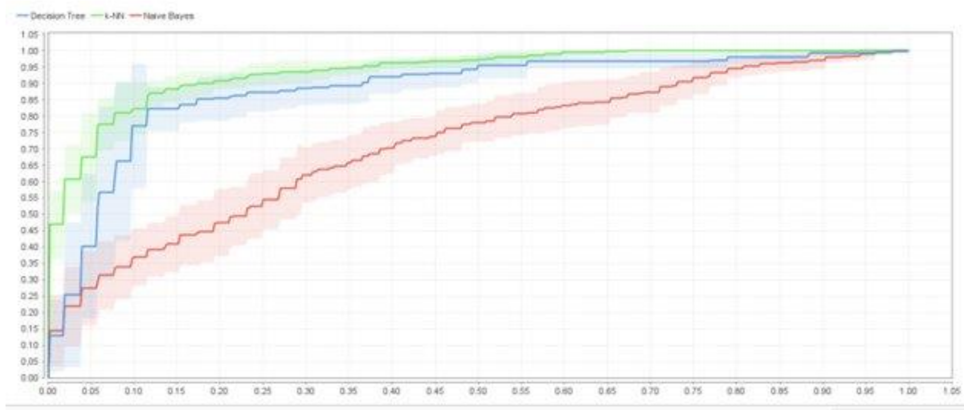
First, we compared the performance of the models when Total Score was the label attribute. To use a ROC comparison, we had to discretize the scores into binary categories for which we chose Fail (<50) and Pass (<100). For this classification, the decision tree model performed the best.



Next, we changed the way that we discretize the total score. We used a Pass (<75) bin and an Excel (<100) bin. For this comparison, the Naïve Bayes model performed the best.



For the comparison of the performance of the models when Gender is the label attribute, we discretized Total Score into five bins using the discretize by binning operator. For this classification, the k-NN model performed the best. This is interesting that the inclusion of the three individual score categories increased the performance of the model.



Conclusions

After completing the classifications, we wanted to address the questions that we asked at the beginning of our project. For the first question, we found that using test preparation does tend to increase scores on the exam and the parent level of education does not influence the test scores. For the second question, we found that most of our classifications were not very accurate, as most of the accuracies were less than 70%. Based on these low accuracies, it would be difficult to determine the performance of an individual student. It could be more accurate with a larger dataset for training. For the third question, we were able to predict gender accurately using the k-NN classification method. It was our most successfully predicted attribute. For the fourth question, we think that prediction of future data would be okay. We would want to be cautious about our predictions since the accuracies of our models averaged less than 70% accurate. Again, we think that with more training data, it would be able to predict more accurately.

For future studies, it would be nice to have information about whether all the students were from one school and the type of school attended. For instance, there could be differences seen amongst different types of schools, such as public vs private and all-girls vs all-boys vs co-ed.