ANALYZING STUDENT PERFORMANCE BY: TIMOTHY WILLISTON

BUSINESS UNDERSTANDING

Student performance is very important in the national, state and local levels. So much of our state government budgets are spent on education. And, for children in primary or secondary school, many hours/days are devoted to education (around 180 days every year).

A students family life has a very big impact on their student performance. Naturally a child's parent(s) is their first teacher (Fairbanks, 2024). I am a parent of four teenagers (at the time that I am writing this). For the last 19+ years, I have been an at-home parent to my kids, so I am very interested in the factors that affect student performance.

So, how do we define student performance, and what do I want to accomplish through this project? Student performance cannot measure or predict if a student is going to be successful in their job/college/life, because there are so many factors that can't be measured. In this analysis student performance will be measured by the grades that they receive in school. When a student as a higher GPA (grade point average), they have more options for college/training, and they have more opportunity to have more scholarships if they attend a university or college.

Student performance is also immensely important to economies and our workforce. "Our nation's economy depends on a well-educated, high-quality workforce and that means investing in all students" (*Research Shows That When It Comes to Student Achievement, Money Matters*, 2018). This applies to the state, local, and neighborhood level as well.

The goals of this analysis will be to look at the factors that affect student performance and find the factors that influence it. I will use real-world census data to analyze this problem, and I hope that this knowledge will inform not only my family but other families, educators, and relevant institutions.

DATA UNDERSTANDING

It is difficult to find data to study this problem. Local or state educational institutions typically don't publish the raw data that they collect on students. But the US Census Bureau does publish some of their raw data. This analysis will be using the 2023 National Survey of Children's Health (NSCH). It can be found on the census.gov website, and it is publicly available for anyone to download. They offer the data for 2016 – 2023 by year, so multi-year

Survey topics include:

- Child and family characteristics
- Physical and mental health status, including current conditions and functional difficulties
- Health insurance status, type, and adequacy
- Access and use of health care services
- Medical, dental, and specialty care needed and received
- Family health and activities
- Impact of child's health on family
- Neighborhood characteristics

Figure 1 – Survey Topics (US Census Bureau, 2025)

analysis is feasible. The data is available in SAS or STATA formats, and there is a useful online Interactive Data Query. Also available is a comprehensive variable list (data dictionary) and frequency/distribution by variable.

This data comes from a household survey and has national and state-level data on the health of children ages 0-17. The survey is completed online, mail, or over the phone, and the questions are asked about a randomly selected child in the household.

The dataset comes with 55,162 records and 456 columns. Many of the variables do not have descriptive name, but codes like the Depression variables (K2Q32A - Fig. 2). This fig. also shows how many variables have multiple features for each. Many of the variables are binary, like Depression. Many others are ordinal like the variable in the lower part of Fig. 2. The ordinal variables will be treated like nominal, but missings won't be imputed.

The data is relatively clean; looking at the ranges for each variable, every number looks good with no outliers. The biggest issue is many missing values. After importing the dataset, I removed columns that had more than 40% missing, and the columns went from 456 to 262.

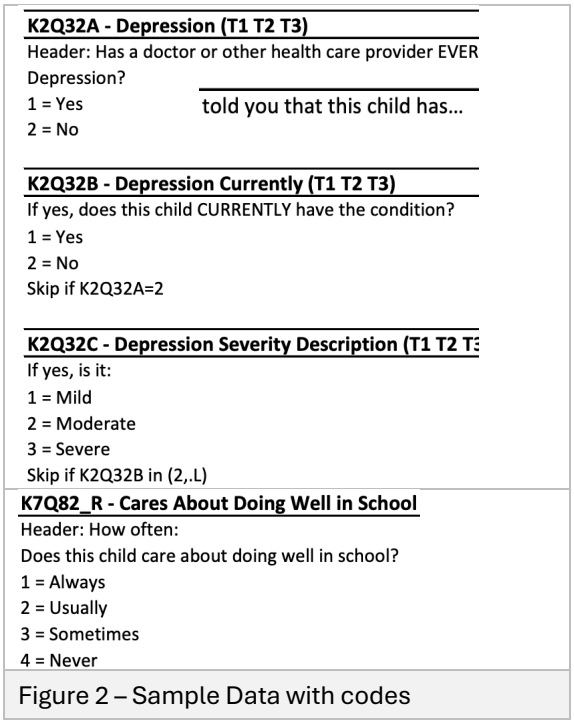


Figure 2 – Sample Data with codes

Fig. 3 shows distributions of the numeric variables and an example of some of the categorical variables. The numeric variables all look relatively normal. A1_Age is the age of the primary caregiver in the home. It has a bump at age 70 because anyone older than 70 is also put there. Birth year has very few younger than 5 years old, but I am going to filter them out anyway.

The categorical examples show one of the issues of those variables. "1" means yes, and "2" means no. These are not well balanced, but that will be helped by a cluster analysis during the data preparation step.

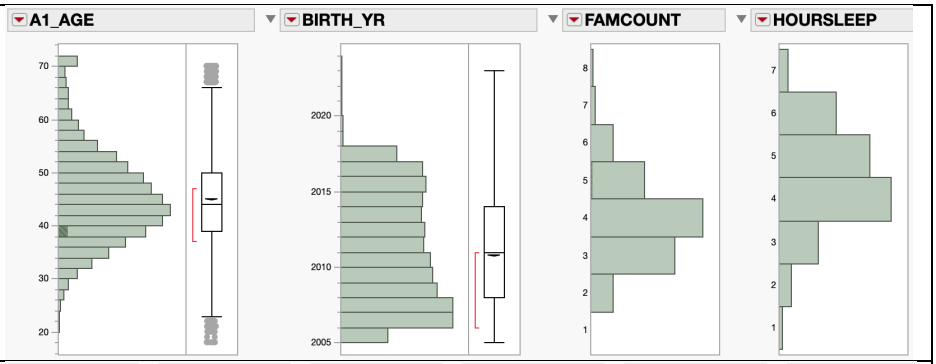
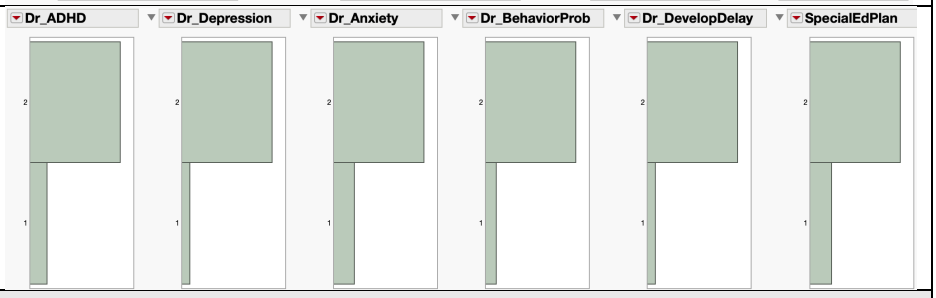
Nominal Variables	
Categorical Variable Example	

Figure 3 – Select Distributions

TARGET VARIABLE

Fig. 4 shows the distribution of the target variable GRADES. The breakdown of the variable is:

1 = Mostly A's
2 = Mostly A's and B's
3 = Mostly B's and C's
4 = Mostly C's and D's
5 = Mostly D's or lower
6 = This child's school does not give these

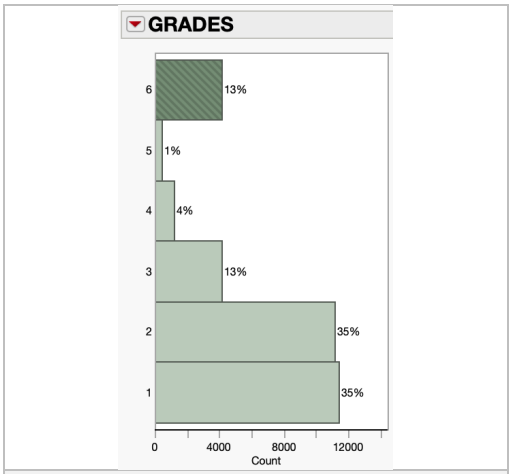


Figure 4 – Grades Distribution (Target)

This is the original distribution of the variable. The (6)s will have to be removed for training the models. The variable will be renamed "HighGrades", and (1) will be "Yes" with (2-5) being "No". The final distribution will be shown in the next step of the analysis.

DATA PREPARATION

Data Preparation for this analysis was done using KNIME and JMP Pro. KNIME is my choice for most of the analysis because the "node" nature makes for a repeatable/editable workflow. For example, this analysis is done using the data from 2023. Later, if newer data becomes available, all I need to do is edit the SAS reader node to point to the new dataset and press "go" to run the entire workflow on the new dataset. But, KNIME doesn't have the Latent Class Analysis feature, so the LCA and some Exploratory Data Analysis will be done in JMP Pro.

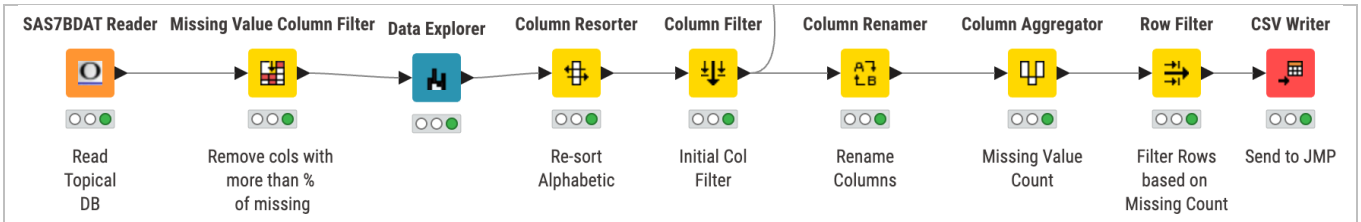


Figure 5 – Initial processing in KNIME

A few steps needed to be done before sending the data to JMP (Fig. 5):

- Read SAS dataset into KNIME
- Remove columns with too many missing values. It was set to 40%
- Explore the data visually – ranges, missings, zeros, and histograms
- Sort the columns alphabetically to match the data dictionary
- Look through the data dictionary and select possible relevant variables
- Rename coded variables to give them meaning
- Count missing values in each row and append to the dataset
- Filter out rows with more than 5 missing values

The dataset started with 55,162 records and 456 columns, and the above process took it to 32,507 records and 38 columns.

LCA AND EDA IN JMP

In JMP, the table view and the header graphs make the data easier for me to see and explore. This view made it easy to see some of the imbalances in the data.

There were several variables, like the ones in Fig. 2, that are like: “Has a Doctor ever told you that your child has…” ADHD, Depression, Anxiety, etc. Most of the answers to each question was “No” (Fig. 6), but I wanted to use them effectively in the analysis despite the imbalanced distributions.

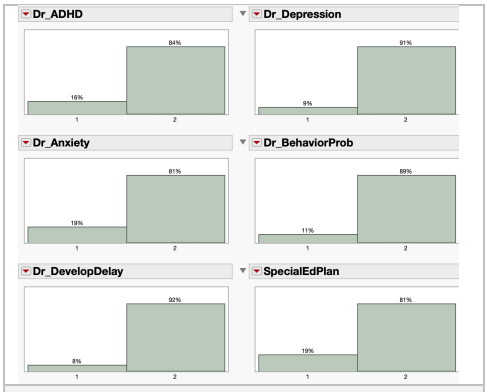


Figure 6 – Distributions of the “Dr” variables

So, a Latent Class Analysis was performed on those variables to combine them into clusters to use for the predictive model. Cluster sizes of 3, 5 and 2 were considered, and 2 clusters were chosen because additional clusters beyond two had very small quantities.

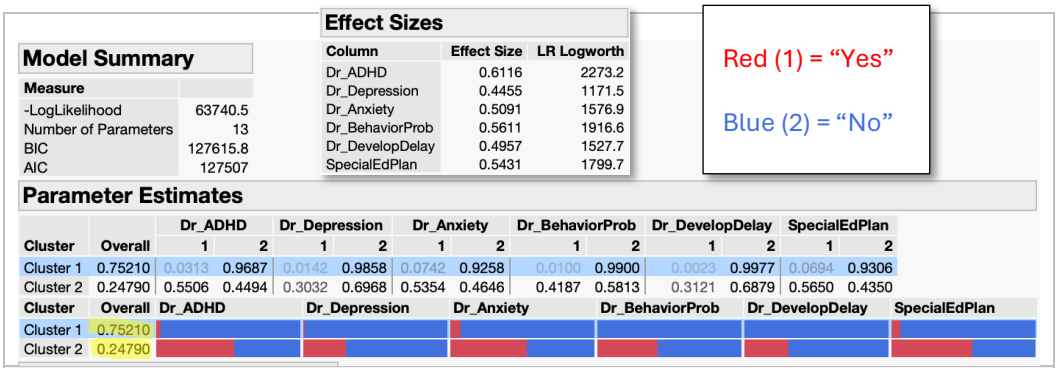


Figure 7 – LCA Results

Fig. 7 shows the results of the LCA analysis. The overall percentages (highlighted) are better than the original splits: 75%/25%. Cluster 1 is mostly children who have not been given a diagnosis by a doctor, and Cluster 2 represents those who have been diagnosed with some of these challenges. The effect size and logworth show that all variables have a similar influence on the cluster outcomes with Dr_ADHD being the highest. After analyzing these results, the cluster id was saved to a new column in the data table, and the probability formula was published/saved to the Formula Depot in JMP to make it easier to id clusters for future use.

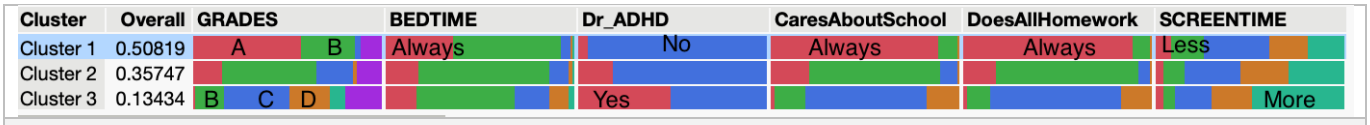


Figure 8 – LCA Secondary Cluster for EDA

The first LCA will be used for our predictive model, but I wanted to do a secondary LCA for EDA purposes. Fig. 8 shows a 3 cluster LDA using six variables that sometimes cause issues in my own home. Cluster 1 is the highest grades; Cluster 2 is in the middle; Cluster 3 is the lowest grades. This is a great way to see correlation among these variables. Cluster 1 (highest grades) contains children with the best bedtime routine, less ADHS, more caring about school, more homework, and less screentime.

DATA PREPARATION IN KNIME

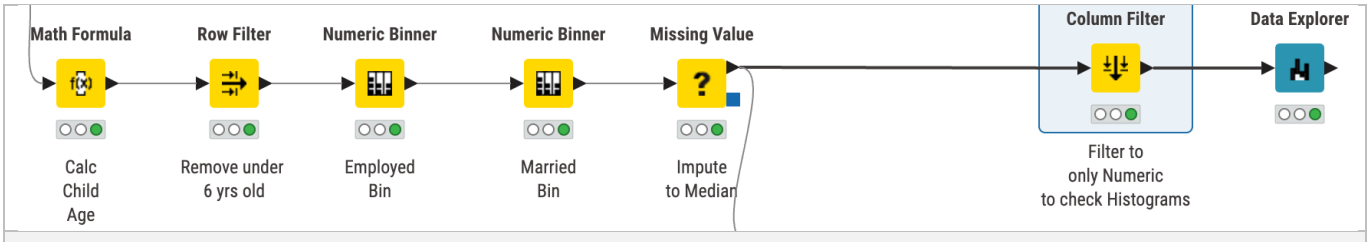


Figure 9 – Continuous Variable Preparation Workflow

After the LCA and visualizing some of the variable distributions, the dataset was exported from JMP and imported back into KNIME as a SAS dataset. Here were the steps for engineering, binning, and imputing the continuous variables (Fig. 9):

- Calculate the age of the child from the birth year

- Remove records for children under 6 years old
- Create a categorical variable from Employed (FullTime, UnderEmployed, Retired)
- Create a categorical variable from Married (Married_trad, NonTrad)
- Impute missings to the Median (A1_AGE, A1_GRADE, BEDTIME, FAMCOUNT, HOURSLEEP)
    - I thought these variables would work well for imputation
    - The others with missings were removed in a later step
- The last two steps were to check histograms to make sure imputations worked well

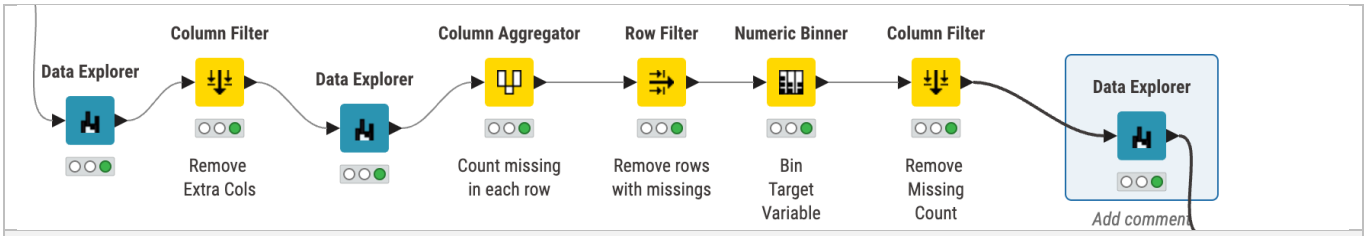


Figure 10 – Categorical Variable Preparation Workflow

There wasn't much data preparation needed for the categorical variables. Most of them were categorized well in the dataset.

SCREENTIME - How Much Time Spent with TV, Cellphone, Computer 1 = Less than 1 hour 2 = 1 hour 3 = 2 hours 4 = 3 hours 5 = 4 or more hours	BEDTIME - How often does this child go to bed at about the same time on weeknights? 1 = Always 2 = Usually 3 = Sometimes 4 = Rarely 5 = Never

Table 1 – Categorical Variable Distribution Examples

Table 1 shows two of the variable distributions. They don't need to be transformed because they already have a good distribution what will work well with the predictor models.

GRADES - Grades Received in School **- HighGrades**

1 = Mostly A's - **Yes**
2 = Mostly A's and B's - **No**
3 = Mostly B's and C's - **No**
4 = Mostly C's and D's - **No**
5 = Mostly D's or lower - **No**

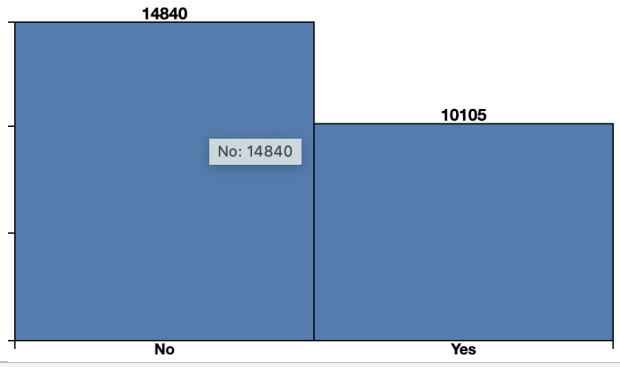


Figure 11 – Target Variable Transformation

Figure 10 shows the steps needed:

- Extra columns were removed after transformations
- Every row left with missing values were removed (29,734 to 24,945 rows)
- The Target variable, GRADES, was transformed to HighGrades (Fig. 11)
- The final Target variable distribution is shown in Fig. 11 also.
- Remove one more column, missing count

At this point the dataset was ready for modeling, which also meant that I can calculate the variable importance and show the final predictor variable list.

The dataset was sent to a *Random Forest Learner* with no partitioning (just for variable importance only). The Random Forest has an output for Attribute Statistics that have the Splits and Candidates. There are three levels to each, and a weighted sum was applied to these to output variable importance.

```
Wsum = (split0/candidate0) + (split1/candidate1)* .1
+ (split2/candidate2)*.01
```

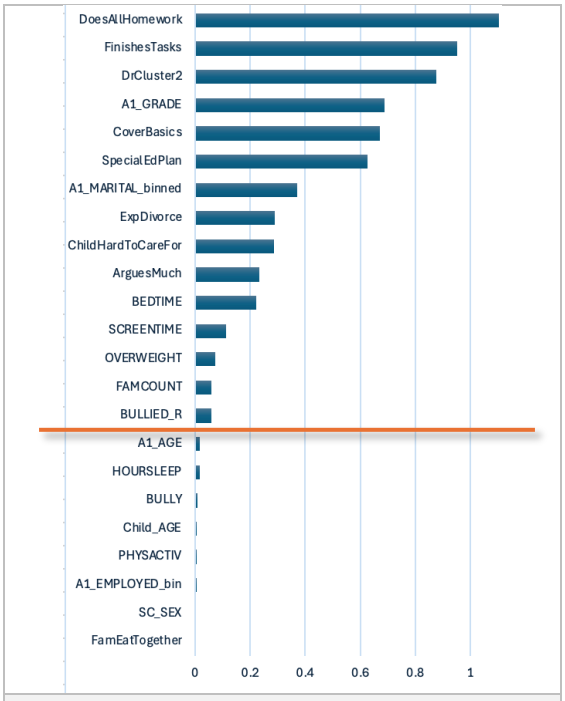


Figure 12 – Variable Importance

The attribute statistics were sent to Excel for more processing. The splits/candidates were tallied in a weighted sum to create the Variable Importance chart in Fig. 12. Based on this output, it was clear that the bottom 8 variables did not have much importance, so these were removed in the final stop of Data Preparation.

MODELING

I chose 6 ML modeling techniques to try for this analysis. The table below will have a breakdown of each model, pros/cons, and chosen parameters.

Decision Tree	Decision Trees are one of my favorite models. They are very transparent and easy to explain. They work well with categorical and numeric variables. But, if not pruned, they tend to overfit and may not be the best fit for the larger dataset that I'm using. My Decision Tree will use the Gini index and MDL pruning method (Reduced Error). It will use 2 minimum records per node and average split point.
Random Forest	Random Forest models always seem to perform well in my experience. Like Decision Trees, they work well will categorical and numeric variables. But, they are a bit harder to visualize and explain. They use lots of memory, but can use parallel processing in a server setting. The Random Forest model will use all the available predictors, with an Information Gain Ratio split criterion. 100 models will be run using "12345" as the seed.

Gradient Boosted Trees	Boosted Trees can be used for regression or classification tasks. It uses many decision trees, and it learns from the errors of the previous trees. It is a black box to me and would be difficult to explain. It is computationally expensive but has given great results in past modeling. The Boosted Tree model will use all the available predictors, and a limit of 4 for tree depth. It will use the default 100 models and 0.1 learning rate.
Artificial Neural Network	The Neural Network (MultiLayerPerceptron) is a fascinating algorithm to me. Seeing a visual representation of one is like art to me. But they are also very flexible and can be used in many ways. They are very customizable, and there are different kinds of Neural Networks. They are modeled after the human brain, and they use weights, bias, and activation functions to train the model to get the best results. But Neural Networks are also a black box, meaning they are difficult to understand and explain. They also consume a lot of computational resources and have been one of the slowest models to run in my experience. KNIME uses a MultiLayerPerceptron learner/predictor. I have had the best results with the default parameters: 100 max iterations, 1 hidden layer, 10 hidden neurons, and a seed of 12345.
Logistic Regression	Logistic Regression is a great algorithm to start with. It's simple and easy to understand and runs quickly. One of the downsides though, is that its simplicity means that it will not find complex patterns in the data (it assumes a log/sigmoid relationship). The LR Learner in KNIME will use Stochastic average gradient and the default "Advanced" parameters: Lazy calculations, calculate coefficients, 100 epochs, Fixed learning strategy, 0.1 step size, and Uniform regularization.
Support Vector Machines	Unlike Logistic Regression, Support Vector Machines are great at finding complex patterns in the data. This makes them useful for a wide range of applications, like image/text recognition, medical classification, time series, and web ranking. They classify data in multiple dimensions using hyperplanes. But they seem to be difficult to tune and find the right kernel parameters. They also don't work well with large datasets, although it worked in my case of 25,000 records with 7 columns. In my case, I will go with the default of 1.0 overlapping penalty and a Polynomial kernel (1.0 for Bias, Power, and Gamma).

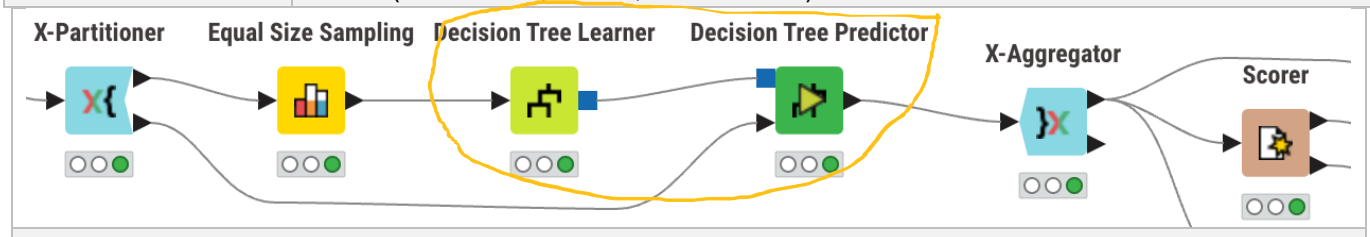


Figure 13 – Sample Model Workflow in KNIME

Fig. 13 shows the modeling workflow. The first three models, Decision Tree, Random Forest, and Gradient Boosted Trees use this workflow. The only difference is the circled nodes. Each use 10-Fold Cross Validation and Equal Size Sampling. Each goes into the Scorer node which will give the confusion matrix and accuracy statistics.

The last three models, Neural Network, Logistic Regression, and Support Vector Machines use the same workflow with two additional steps before them; the data is One-Hot Encoded; that is the non-numeric categorical variables are put into separate columns with 1 or 0. Then they are normalized; all variables are transformed to between 0 and 1 to keep any large-scale variables from dominating. Also, the SVM uses a smaller subset of data to allow it to run on my computer (top 5 predictors from Fig. 12).

EVALUATION

Each model outputs the prediction table (X-Aggregator) to an appender node. This is used to make the combined ROC Curve graph (Fig. 14) to help decide which model we want to use. Plotting the True Positives vs the False Positives, this area under the curve is a great metric for model performance. In this case, the Boosted Trees, Neural Network, and Logistic Regression tied for highest. Also, it's good to see tha AUC over 0.8, which is a good indicator that our model is performing well.

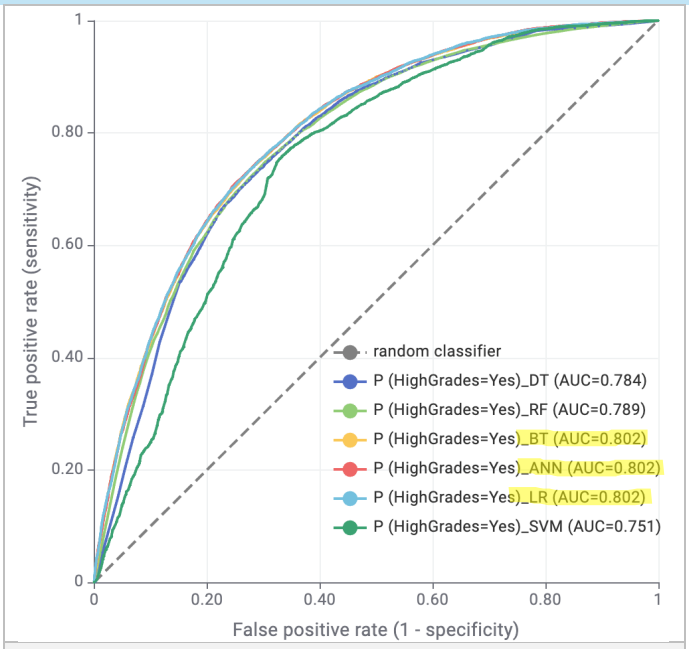


Figure 14 – Combined ROC

The scorer node from Fig. 13 has two outputs. The top is the Confusion Matrix, and the bottom is the Accuracy Statistics. These were aggregated and concatenated and written to an Excel spreadsheet to maintain number accuracy and for repeatability.

		Yes	No	Yes	No	Recall	Precision	Sensitivity	Specificity	Accuracy
Decision Tree	Yes	7472	2633	30.0%	10.6%	73.9%	62.7%	73.9%	70.0%	71.6%
	No	4446	10394	17.8%	41.7%					
Random Forest	Yes	7702	2403	30.9%	9.6%	76.2%	61.9%	76.2%	68.0%	71.3%
	No	4746	10094	19.0%	40.5%					
Gradient Boosted Trees	Yes	7536	2569	30.2%	10.3%	74.6%	**63.7%**	74.6%	**71.1%**	**72.5%**
	No	4295	10545	17.2%	**42.3%**					
Artificial Neural Network	Yes	7573	2532	30.4%	10.2%	74.9%	**63.7%**	74.9%	70.9%	**72.5%**
	No	4321	10519	17.3%	42.2%					
Logistic Regression	Yes	7642	2463	30.6%	9.9%	75.6%	63.2%	75.6%	70.0%	72.3%
	No	4445	10395	17.8%	41.7%					
Support Vector Machines	Yes	7882	2223	**31.6%**	8.9%	**78.0%**	59.6%	**78.0%**	64.0%	69.7%
	No	5338	9502	21.4%	38.1%					

Figure 15 – Accuracy Statistics and Confusion Matrices

Fig. 15 shows the Confusion Matrices (left side) and the accuracy statistics. We are trying to predict HighGrades from various predictors, and the positive class value is what we want to look at first (True Positive Rate – Sensitivity). The SVM wins here (78.0%), but there is more to the story.

Overall Accuracy and the Sensitivity/Specificity ratio are also important. Boosted Trees and the Neural Network both have the same overall accuracy, but Boosted Trees has only a 3.5% difference between Sensitivity and Specificity. Therefore, Boosted Trees is our model champion!

For the sake of curiosity, I exported the final dataset back to JMP to run a model screening. JMP would not run a SVM because it had too many records, but Fig. 16 shows the result. JMP picked the Neural Network (Neural Boosted) and Boosted Tree as the top two.

Summary Across the Folds

			Validation Set Folds				
Method	N Trials Folds	Sum Freq	RSquare	Mean RASE	StdDev RASE	Mean AUC	Mean MR
Neural Boosted	10	2494.5	0.2227	0.41949	0.00368	0.8049	0.26354
Nominal Logistic	10	2494.5	0.2182	0.42074	0.00381	0.8026	0.26378
Boosted Tree	10	2494.5	0.2168	0.42114	0.00351	0.8019	0.26350
Bootstrap Forest	10	2494.5	0.2116	0.42282	0.00309	0.7986	0.26703
Decision Tree	10	2494.5	0.2102	0.42329	0.00339	0.7963	0.26811

Figure 16 – JMP Model Screening

DEPLOYMENT

In KNIME we can easily deploy this model in a few ways. First, the whole workflow and be re-run quickly with new data. Also, new data can be connected directly to the "Learner" node to predict new data. Lastly, the "Learner" node can be exported to PMML (Predictive Model Markup Language) and sent to others to use in other software if desired.

Besides that, I hope this analysis is educational to my family and others. It could be used by non-profit institutions to help know what predicts student performance the most. Student performance is truly important on the family, neighborhood, town, state, and national level. Doing well in school can change the trajectory of a family, and it can benefit children for their whole lives.

REFERENCES

Fairbanks, C. C. (2024, April 15). *The influence of family on student outcomes*. Sutherland Institute. https://sutherlandinstitute.org/the-influence-of-family-on-student-outcomes/

Research Shows That When it Comes to Student Achievement, Money Matters. (2018, July 17). Learning Policy Institute. Retrieved April 28, 2025, from https://learningpolicyinstitute.org/press-release/research-shows-student-achievement-money-matters

US Census Bureau. (2025, April 8). *About the National Survey of Children's Health*. Census.gov. https://www.census.gov/programs-surveys/nsch/about.html

APPENDIX - DATA DICTIONARY FOR RELEVANT VARIABLES

A1_AGE - Adult 1 - Age in Years (T1 T2 T3)
What is your age? [18-70 or older]

A1_BORN - Adult 1 - Where Born (T1 T2 T3) Where were you born?
1 = In the United States
2 = Outside of the United States

A1_EMPLOYED_R - Adult 1 - Current Employment Status (T1 T2 T3)
1 = Employed full-time
2 = Employed part-time
3 = Working WITHOUT pay
4 = Not employed but looking for work
5 = Not employed and not looking for work
6 = Retired

A1_GRADE - Adult 1 - Highest Completed Year of School (T1 T2 T3) What is the highest grade or level of school you have completed?
1 = 8th grade or less
2 = 9th-12th grade; No diploma
3 = High School Graduate or GED Completed
4 = Completed a vocational, trade, or business school program
5 = Some College Credit, but No Degree
6 = Associate Degree (AA, AS)
7 = Bachelor's Degree (BA, BS, AB)
8 = Master's Degree (MA, MS, MSW, MBA)
9 = Doctorate (PhD, EdD) or Professional Degree (MD, DDS, DVM, JD)

A1_MARITAL - Adult 1 - Marital Status (T1 T2 T3) What is your marital status?
1 = Married
2 = Not married, but living with a partner
3 = Never Married
4 = Divorced
5 = Separated
6 = Widowed

ACE1 (CoverBasics) Hard to Cover Basics Like Food or Housing
1 = Never
2 = Rarely
3 = Somewhat often
4 = Very often

ACE3 (ExpDivorce) - Child Experienced - Parent or guardian divorced or separated
1 = Yes, 2 = No

BEDTIME - How often does this child go to bed at about the same time on weeknights?
1 = Always
2 = Usually
3 = Sometimes
4 = Rarely
5 = Never

BULLIED_R - Bullied, Picked On, or Excluded by Others (T2 T3) DURING THE PAST 12 MONTHS,
1 = Never (in the past 12 months)
2 = 1-2 times (in the past 12 months)
3 = 1-2 times per month
4 = 1-2 times per week
5 = Almost every day

BULLY - Bullies Others, Picks on Them, or Excludes Them
1 = Never (in the past 12 months)
2 = 1-2 times (in the past 12 months)
3 = 1-2 times per month
4 = 1-2 times per week
5 = Almost every day

FAMCOUNT - Number of People That Are Family Members
[1-8 or more]

GRADES - Grades Received in School
1 = Mostly A's
2 = Mostly A's and B's
3 = Mostly B's and C's
4 = Mostly C's and D's
5 = Mostly D's or lower
6 = This child's school does not give these grades

HOURSLEEP - Past Week - How Many Hours of Sleep Average
1 = Less than 6 hours
2 = 6 hours
3 = 7 hours
4 = 8 hours
5 = 9 hours
6 = 10 hours
7 = 11 or more hours

Dr_ADHD – Has a Dr or health care provider EVER told you that this child has...
1 = Yes
2 = No

Dr_Depression – Has a Dr or health care provider EVER told you that this child has...
1 = Yes
2 = No

Dr_Anxiety – Has a Dr or health care provider EVER told you that this child has...
1 = Yes
2 = No

Dr_BehaviorProb – Has a Dr or health care provider EVER told you that this child has...
1 = Yes
2 = No

Dr_Autism – Has a Dr or health care provider EVER told you that this child has...
1 = Yes
2 = No

Dr_DevelopDelay – Has a Dr or health care provider EVER told you that this child has...
- 1 = Yes
- 2 = No

Dr_IntellDisability – Has a Dr or health care provider EVER told you that this child has...
- 1 = Yes
- 2 = No

SpecialEdPlan - Has this child EVER had a special education or early intervention plan?
- 1 = Yes
- 2 = No

ArguesMuch - Does this child argue too much?
- 1 = Always
- 2 = Usually
- 3 = Sometimes
- 4 = Never

CaresAboutSchool - Does this child care about doing well in school?
- 1 = Always
- 2 = Usually
- 3 = Sometimes
- 4 = Never

DoesAllHomework - Does this child do all required homework?
- 1 = Always
- 2 = Usually
- 3 = Sometimes
- 4 = Never

FinishesTasks - Works to Finish Tasks Started
- 1 = Always
- 2 = Usually
- 3 = Sometimes
- 4 = Never

FamEatTogether - How Many Days - Family Eat Meal Together
- 1 = 0 days
- 2 = 1-3 days
- 3 = 4-6 days
- 4 = Every day

ChildHardToCareFor - Child Hard to Care For
- 1 = Never
- 2 = Rarely
- 3 = Sometimes
- 4 = Usually
- 5 = Always

OVERWEIGHT - Doctor Identified as Overweight
- 1 = Yes
- 2 = No

PHYSACTIV -
Exercise, Play Sport, or Physical Activity for 60 Minutes
- 1 = 0 days
- 2 = 1 - 3 days
- 3 = 4 - 6 days
- 4 = Every day

SC_SEX (GENDER) –
- 1 = Male
- 2 = Female

SCREENTIME - How Much Time Spent with TV, Cellphone, Computer
- 1 = Less than 1 hour
- 2 = 1 hour
- 3 = 2 hours
- 4 = 3 hours
- 5 = 4 or more hours