MVESC-DSSG Technical Plan

Identifying and Influencing Students At Risk of Not Finishing High School

Updated: August 22, 2016

Partner: Muskingum Valley Educational Service Center
Project Manager: Chad Kenney
Technical Mentors: Ali Vanderveld, Kevin Wilson
Fellows: Xiang Cheng, Jacqueline Gutman, Johanna Torrence, Zhe Zhang

I. Problem Formulation

About 1 in 5 students across the U.S. do not graduate high school on-time or do not finish high school. Currently, many intervention and support systems have the problem of "too little, too late". One, schools and teachers have limited data available, especially across districts. Two, red flags to identify students who are at-risk tend to occur mostly after already extreme outcomes — such as enough absences to flag a truancy issue or many failed classes — when it may be more difficult for interventions to put a student back on-track. Identifying these students at risk of future delayed or non-graduation can help schools intervene and support these students earlier, when interventions may be more impactful.

We are working with an engaged and helpful partner for this problem, Muskingum Valley Educational Service Center (MVESC). Importantly, they work with 16 partner school districts and in the last decade, have collected a dataset that is both comprehensive across their districts and rich in longitudinal and detailed information (e.g. teacher information, daily absences, disability accommodations, and extracurricular activities). They want to use the information here to create a better early warning system for at-risk student, as early as it can be effective.

A useful final product is a ranking of current students in various grade levels who are at risk, their current risk level, and importantly, interpretable insights into why each individual student has been deemed at risk.

To this aim DSSG will help MVESC with three specific problems:

- **Problem 1: An "early warning system" to predict which students are at risk of not finishing high school, at different times in a student's career.**

- **Problem 2: An analysis of the tradeoff between earlier prediction and more accurate prediction.**
- **Problem 3: A spreadsheet of student risk for individual schools, with interpretable risk factors for each individual student too.**

II. Data Summary

Data Overview

We have been given various primary datasets by MVESC which will be useful in our problem. Between districts, the amount of years of data available vary.
(See https://docs.google.com/a/uchicago.edu/spreadsheets/d/1r6pp6_4G67kkpgSOX9PxXNyH4stscv77DRxYtGbM8og/edit?usp=sharing for a more detailed description of data availability, though it needs updating. Information on IEP accommodations, teachers, and interventions have not been detailed there)

Dataset	**Variables**	**Aggregation**
Absences	aggregated and daily absence/tardy, reason	student, year, district
Yearly Demographics & Snapshot	Age, gender, language, ethnic., address, learning or educational disability	student, year, district
Class Grades	Mark received per class, class name, and length of class	student, term (or year), district
Ohio Assessment and Graduation Test Scores	Grades 3, 4, 5, 7, 8 standardized test scores; high school graduation test scores	student, year, district
Other Test Scores	STAR English (K-2), STAR Math (grades 1-10), STAR Reading (grades 1-10)	student, grade level, district
IEP Accommodation Details (for 3 districts only)	Grade of test for accommodation, specific test type, free text description of accommodations for each test	student, test grade level, district
Teacher/student pairings	Year, teacher code, course name, course type	student, year, district
Intervention/Membership Indicators	Year, grade level, indication of if a student received an intervention / extracurricular activity	student, grade, year, district

Data Storage

We use a PostgreSQL database to store the database backups, CSV files, and Excel files that DSSG received.

Data Pipeline

See the Github ReadME files for more details on each section. These have been documented at the end of our summer project (~Aug 19 2016).

/ETL

- Database loading of raw data (Python)
- Cleaning and standardizing of raw data (SQL, Python)

/Descriptives

- Data exploration & data stories (Python)

/Features

- Skeleton model pipeline & feature generation (Python, SQL)
- Feature generation into `model` schema (Python, SQL)

/Models_Results

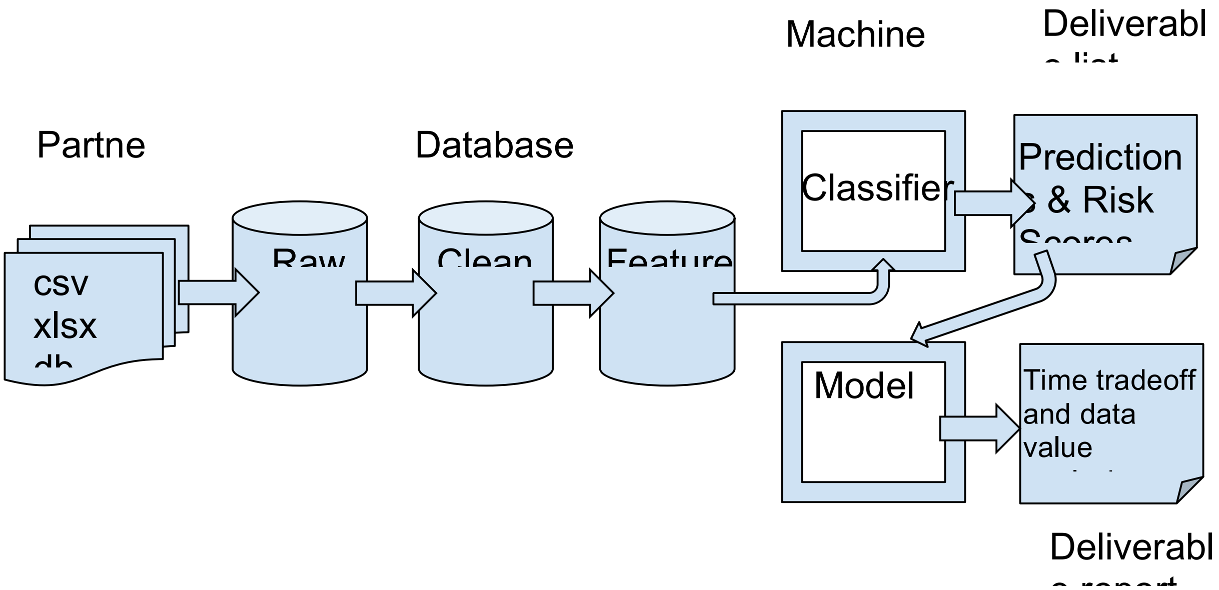
- Reporting and saving module to pipeline (Python)
- Parameter grid search methods for various models (Python)
- Missing data imputation (scikit-learn)
- Create balanced training data with bootstrapping or downsampling (?)
- Functions to automatically loop through meta-parameters such as feature sets, time of prediction, cross-validation scheme, downsampling weight, etc; creating a YAML file for each to then execute a model estimation. (Python)
- Write some summary results and predictions to the database, as well as to a Markdown report (Python)

/Reports

- Code to perform some visualizations and analysis of various sets of model runs (Python, R)

/Error_Feature_Analysis

- Analysis of feature importance and model errors



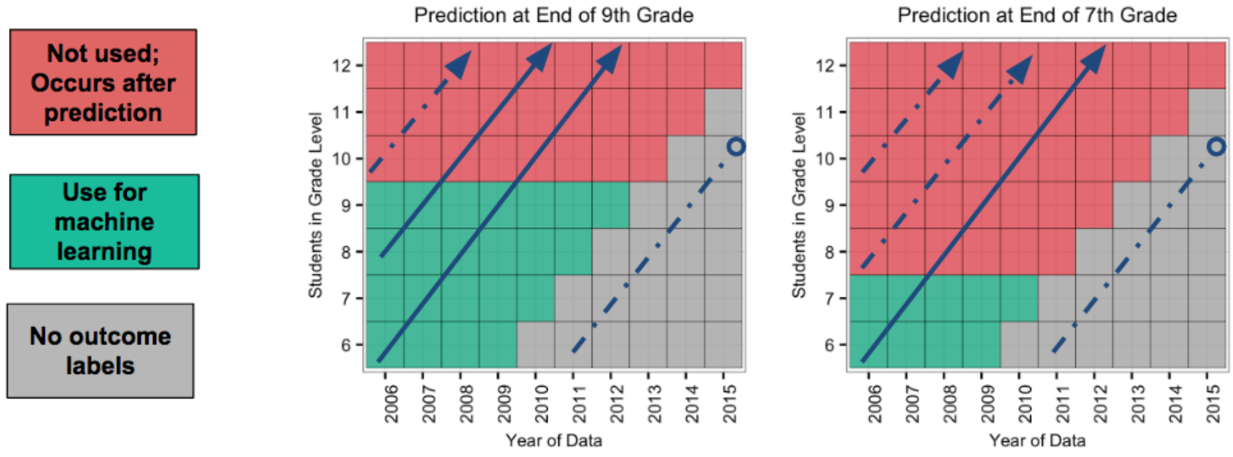
III. Analytical Approach

Problem 1: An “early warning system” to predict which students are at risk of not finishing high school.

To address this problem, we had three main tasks:
(a) creating a training set by identifying the outcomes of students in our data,
(b) creating a wide set of grade level-specific features, and
(c) training a predictive classifier, at different grade levels, to predict which students are at-risk.

This allows us to predict at various times or grade levels in a student’s career. With our different predictions, we can use the results to study Problem 2.

Subproblem A — Identifying Labels



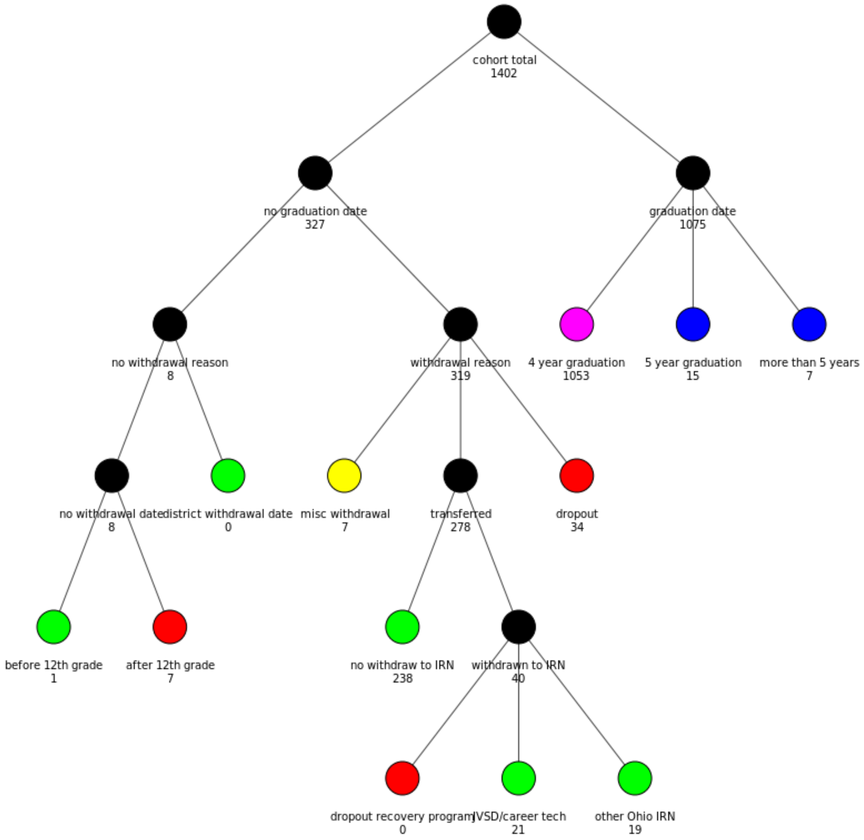
The above graphic visualizes the data that we have available to us. Each column in the graphic is a yearly snapshot of all students in a given year across all grades, but we don’t have data going back in their past (with the exception of standardized test scores). Using these snapshots from 2006/07 to 2015/16 school years, we can thus track individual student ID numbers across time, and further track *cohorts* across time.

The most recent cohorts, marked in gray on the graphic, we cannot use for training our machine learning models. This is because for the (vast majority of) students in those cohorts, we have not yet observed their graduation outcomes, so we cannot use them as labeled data.

Similarly, the earliest cohorts — e.g. the cohort of students who begin 10th grade in 2006-07 — cannot be used for training. While we observe their outcomes, we don’t observe any data before 10th grade that we can use for training. These cohorts are marked in red in the “Prediction at End of 9th Grade” graphic (code for making this graphic in Github). This leaves the data marked in green as available to us for machine learning. (The second graphic, to its right, is included to show how performing prediction at an earlier grade level (beginning of 8th grade) limits the available data.)

Finally, as an accounting measure, it’s important to note how to define a cohort. We identify which year a student in Grade X and mark that year as their Grade X cohort year. However, a student could be in multiple Grade X cohort years, for example, if they repeat Grade X. We are okay with this because we are predicting at the beginning of Grade X, so this student would be given a risk score using the Grade X model each year that they are in Grade X.

Within our available data, labels of student outcomes that we can use for training are not obvious. We have documented all the possible outcomes we observe from a student. Please see that documentation for more info. Briefly, for any given cohort of students, we observe the following "outcome buckets": graduated on-time / early / in 5+ years, has a withdrawal date and reason, or has no withdrawal date, instead just disappearing from the data. Here is an example tree for a cohort (useful for the idea, though the text is a bit small):



From this category, we always remove the students with miscellaneous withdrawal codes (e.g. death). Just using these categories there are 3 ways to define our outcomes that require no further choices:

- On-time graduates vs everyone else
- Everyone else vs not-on-time graduates
- Definite outcomes (any-time graduates vs dropouts) — ignoring transfers and uncertain outcomes

The difficulty with these three sets of outcomes is that they either ignore a lot of students, or they group a lot of possibly quite different students together.

For example, in the definite outcomes, we ignore a large portion of students who withdraw. This is important to note because many students who do not graduate high school, may actually be marked as withdrawals by schools. Based on discussions with our partner, schools have an incentive to not mark dropout students due to funding and perception. Further, students often may go or be assigned to "dropout recovery" schools or programs. These have very low actual graduation rates.

Thus, we looked at the uncertain cases and came up with some rough rules-of-thumb to categorize uncertain students. The visuals used here are in /Descriptives/outcome_analysis. We came up with a script to make this categorization in /Features/add_new_outcome_on_ogt.sql. If an uncertain student does not score at least 400 (the passing score) on at least 2 out of 5 graduation tests (conditioned that we observe at least one score) OR they have > 15 absences in Grade 11 or 12 OR they have GPA < 2.0 in Grade 10, 11, or 12; then we mark the uncertain student as a dropout. We also include students who transferred to a joint vocational school (JVSD) or career technical school as a dropout. The other students, who may be legitimately have transferred, we ignore in the training set.

Possible Future Improvements:

It's possible that, in future versions of this, this labeling of students will be easier. First, the schools themselves may have been keeping better track of this information withdrawal reason and IRN codes in recent years. MVESC can highlight the importance of getting this information to build a more accurate model. Second, there is additional auxiliary information that we didn't use: college clearinghouse data, for example, could signal that an 'uncertain' student graduated. Finally, if we believe that uncertain students are different than those who didn't graduate, we could build a multi-label classifier instead of our current binary classifier.

Finally, we could use a semi-supervised approach to help label the uncertain students. Using the students with known outcomes, we could build a classifier on the `definite` students and then apply that classifier onto the uncertain students. To evaluate if this approach is working, we can compare our `definite_plus_ogt` outcome performance vs the `semi-supervised approach` on a held-out validation set.

Subproblem B — Creating Grade-Specific Predictive Features

After identifying student outcomes, we want to create a set of observable features of the students that may be predictive of those outcomes. Broadly, the features we used fall into the following categories, each of which has a **corresponding** table in our `model` schema:

- Absences
- Demographics (collapsed to be grade independent)
- Grades
- Intervention / Extracurricular Information
- Mobility
- Ohio Achievement Assessment state standardized tests
- Yearly Snapshot Information

Further, this Summer 2016, we pursued a wide set of features, but there are still several categories of features that we think would be useful for future work. We discuss these after listing the features we did create.

All of the code to create these features and tables in the `model` schema are in the Github `/Features` folder. Thus, below, we briefly list each feature type, any notable idiosyncrasies and calculation methodology, and refer the reader to the code for the exact specifics.

1. Absences (obtained from the daily attendance data for each year from 2009 - present, for a handful of schools. There is more data available recently, older data is more sparse.)
    a. Total absences in grade X
    b. Unexcused absences in grade X
    c. Total marked tardies in grade X
    d. Unexcused tardies in grade X
    e. Medical absences/tardies in grade X
    f. Number of consecutive absences in grade X

i. Consecutive absence = 2 days in a row, so 3 days in a row is counted as 2 consecutive absences instead of just 1.

g. Number of consecutive tardies in grade X
h. Total absences per day of week in grade X (MTWThF)
i. Total tardies per day of week in grade X (MTWThF)

2. Demographics (collapsed to have only one record per student ID and collapsed to take the most recent observed value, so no grade-level specific information)
   a. Ethnicity
   b. Gender
3. Grades (obtained from yearly class grade data, from 2006 - present for lots of districts)
   a. Language classes GPA in grade X
   b. Num language classes in grade X
   c. STEM classes GPA and number in grade X
   d. Humanities classes GPA and number in grade X
   e. Art classes GPA and number in grade X
   f. Health classes GPA and number in grade X
   g. Future Prep classes GPA and number in grade X
   h. Interventions classes GPA and number in grade X
   i. Overall GPA in grade X
   j. Percent passed of pass/fail classes and number of pass/fail classes in grade X
   k. Overall GPA normalized by district in grade X
4. Intervention / Membership(received later on in the summer, 2006-2016)
   a. Special instruction in grade X
   b. Title I in grade X
   c. Post secondary in grade X
   d. Academic Intervention in grade X
   e. Academic Intracurricular in grade X
   f. Athletics in grade X
   g. Extracurricular program in grade X
   h. School program in grade X
   i. Placement in grade X
   j. Vocational activity in grade X
5. Mobility (obtained from analyzing the `clean.all_snapshots` table for changes and patterns)
   a. Street address changed in grade X from previous year
   b. District changed in grade X from previous year
   c. City changed in grade X from previous year
   d. Mid-year withdrawal in grade X
   e. Num addresses observed up through grade X
   f. Num districts observed up through grade X
   g. Num cities observed up through grade X
   h. Num records observed up through grade X
   i. Avg address changes per year for student up through grade X
   j. Avg district changes per year for student up through grade X
   k. Avg city changes per year for student up through grade X
6. OAA Test Scores. This required first identifying which year each student was in a specific grade, then normalizing each student's test based on all the other test scores in that year. Percentiles were calculated similarly. There are additional tests available in the `clean.oaaogt` table, but some tests were only given at different points in time. Referencing THIS code, we only kept the tests where the test and validation cohorts took those tests — otherwise, we ignored those tests.

    This is why we have reading and math for all grade levels, but only science and/or social studies for a couple. In our best-performing models we included only normalized scores.
    a. Reading score normalized and percentile, based on that school year, using the observed scores in the year for grades 3 - 8
    b. Math score normalized and percentile, based on that school year, using the observed scores in the year for grades 3 - 8
    c. Social studies score normalized and percentile, based on that school year, using the observed scores in the year for grades 5
    d. Science score normalized and percentile, based on that school year, using the observed scores in the year for grades 5 and 8
    e. Placement categories (e.g. Basic, Advanced, etc.) based on raw score for the above reading, math, social studies, and science tests.

7. Snapshots. This is status and descriptive information about a student pertaining to each yearly snapshot of them. (Obtained from the `clean.all_snapshots` table)
    a. Total days absent in grade X
    b. Total days excused absences in grade X
    c. Total days unexcused absences in grade X
    d. Total days present in grade X
    e. Disability status in grade X
        i. a categorical variable of the type of learning or physical disability a student has
    f. Economic disadvantagement status in grade X
    g. Number of discipline incidents in grade X
    h. District name in grade X
    i. Gifted status in grade X
    j. ISS (in-school suspension) count in grade X
    k. Limited english status in grade X
    l. OSS (out-of-school suspension) count in grade X
    m. Section 504 plan status in grade X
        i. Related to an individual student's disability plan
    n. Special ed status in grade X
        i. % of time student participates in special education that year
    o. Student status in grade X
        i. (e.g. active, inactive, local vocational school, ?)

Additional information we not currently using:

1. *district_admit_date**: raw date of a student's entry date into a school district, useful for a creating the features below
2. *street/street2**: a student's home address for a school year
3. *zip**: a categorical value for a student's zip code for a school year
4. *absence_length**: the length of a student's absence or tardy
5. *{kral, kral_pl}:* a numeric kindergarten reading assessment test score and the subsequent ordered categorical value of grouped performance
6. *air_test_score:* (not useful for prediction, only given recently in 2015-16)
7. *asq_preschool_score:* a preschool ages and stages assessment
8. *dibels_v2_score:* early literacy exam no longer given
9. *parcc_test_score:* only given in 2014-15

10. *star_english_score:* this is a set of numeric scores for a student and grade level when they took the Ohio Star English Language test
11. *star_math_score:* this is a set of numeric scores for a student and grade level when they took the Ohio Star Math test
12. *star_read_score:* this is a set of numeric scores for a student and grade level when they took the Ohio Star Reading test
13. *terra_nova_score:* this is only valid for predicting after 9th grade because it was first given in May 2013 (the 2012 cohort of 9th graders)

Features Not Yet Built:

1. *age_in_months*: derived from date of birth of student
2. *top_20percent_age*: indicator if a student is in the top 20% of their cohort in age
3. *bot_20percent_age*: indicator if a student is in the bottom 20% of their cohort in age
4. *grade_enter_district*: set of indicators for the grade a student entered their current district estimated based on *district_admit_date*
5. *continuous_time_in_district:* the number of years a student has been in the school district up to their current grade we are looking at / predicting at, using *district_admit_date*
6. *elementary/middle/high_school:* Using the yearly snapshots, we will identify the name of the school that a student attended for elementary, middle, and high school and place them in a column named like *elementary_school*. The data comes from the *school_name* raw data. If a student attends multiple schools for each section, we will mark it as multiple.
7. *absences_per_month_X*: for each month (e.g. Aug to June) in our data, a count of the number of absences each student has for that month in the school year
8. *absences_per_week_X*: for each calendar week in our data, a count of the number of absences each student has for that week in the school year
9. *consecutive_3plus_absences:* a count for the number of blocks of three absences in a row in the school year
10. *consecutive_4plus_absences:* a count for the number of blocks of four absences in a row in the school year
11. Aggregate/subset absence features by:
    a. *Student Grade Year*
    b. *Multiple Grade-Years grouped together*
    c. *Excused vs Unexcused vs Medical*
    d. *Full absences vs half-day absences vs tardies*
12. *absence_trend_up_over_grade_X:* an indicator if a student's absence frequency notably trended upwards in grade X
13. *absence_variance_in_grade_X:* a measure of a student's variance across the school year in total absences/tardies per week; a low variance signals a consistent pattern across the year, while a high variance signals different school periods with high and low absence frequency
14. *absence_fall*
15. *absence_winter*
16. *absence_spring*
17. *distance_to_school:* the distance of a student's home address to the address of the school they attend in a school year
18. *census_location_income:* the census reported income for a student's address location (inflation adjusted)
19. *marks_trend_down_over_grade_X_Y:* an indicator if a student's marks trend down between the beginning of grade X and the end of grade Y
20. *did_student_repeat_grade_X:*

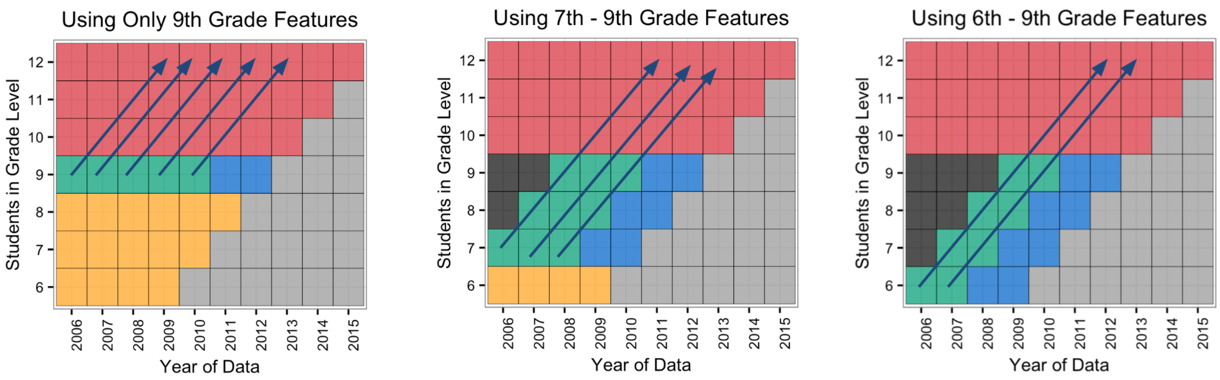
21. *did_student_change_schools_abnormally:*
22. *did_student_performance_fall_after_change:*
23. *number_of_abnormal_school_changes:*
24. *disability_change*: going from no disability to an identified disability (inspired by OSU's work)
25. *oaa_elementary_trend*: a categorical variable characterizing how a student's oaa scores changed across 3rd through 5th grade
26. *oaa_diff_sixth_vs_elementary*: a numeric measuring the change in oaa score between sixth grade tests and elementary school tests, measuring the difficulty of moving to middle school
27. *oaa_diff_gradeX_vs_gradeY*: a numeric measuring the change in oaa score between two grade levels, e.g. 8th grade score minus 6th grade scores
28. *oaa_test_variance_elementary*:
29. *oaa_test_variance_middle_school*:
30. *oaa_missing*: a count of the cumulative number of missing OAA tests for a student
31. *oaa_max_consecutive_neg:* max number of consecutive decreases in normalized oaa score
32. *relative_to_classroom_peers*: compare scores and grades amongst classroom peers
33. *relative_to_teacher:* compare scores and grades amongst other students who had the same teacher
34. *change_address_recent_two_years*: an indicator is a student's home address has changed in the recent two years (currently have change since the previous year)
35. *change_in_disability:* has there been a change in the disability categorical value
36. *change_in_disadvantage:* has there been a change in the disadvantagement categorical value
37. *ACS_income_home*: American Community Survey (ACS) income data by census tract, linked based on geocoded addresses.
38. *ACS_income_school*: American Community Survey (ACS) income data by census tract, linked based on geocoded addresses.
39. *ACS_income_home_school_difference*: American Community Survey (ACS) income data by census tract, linked based on geocoded addresses.
40. *classroom_size_grade_X*: Using the teacher data, a count for the estimated class size for a student in grade X
41. *Teacher_code*:
42. *testing_accommodation_grade_X:* a set of indicator variables noting what testing accommodations a student received in grade X (we only have data for 3 districts, that is the partial reason why we did not use it.)

Subproblem C — Training Predictive Classifiers at Different Grade Levels

Using machine learning to create a risk system for students, we have several "meta-parameter" choices to make:

- Time of prediction (beginning of grade X)
    - As we discussed above, this affects the amount of training data we have available.
- How far back in student history to use before time of prediction
    - This is the second important choice after the time of prediction. This is visualized in the graphic below. The red and grey areas are the same as the graphic above for time of prediction, but within the green area we have added additional colors.
    - On the far left, we choose to only use 9th grade data to predict at the beginning of 10th grade. This means we ignore the data in yellow, and we have 7 total cohorts we can use — denoted by the parallel diagonal arrows.

        - We choose the most recent two cohorts (those that graduated on time in Spring 2015 and Spring 2016) as our Validation set and Test set.
          The **Validation Set** is used for choosing the best meta-parameters, discussed here.
          The **Test Set** is used for a final estimate of our actual performance. It also helps add robustness so that all of our choices are not made on just one cohort.
        - The rest of the cohorts are used for training the classifiers and for parameter estimation. These are colored in green.
    - However, we may want to use data further back in a student's history. Thus, in the middle graphic, we use 7th, 8th, and 9th grade features. If we do this, this limits the number of cohorts we have — going from 7 to 5 cohorts. This is because for the earliest 2 cohorts we don't have 7th grade data, making them unusable for machine learning using 7th - 9th grade features.
    - This tradeoff occurs again in the far right graphic, where getting another year of student history reduces the cohorts available by one.
    - This occurs because of our data is built on snapshots of years, grades, absences, etc going back 10 years. There are exceptions for the standardized test data though, where we have some historical data for all the cohorts going back to 3rd grade.



- Choice of parameter estimation technique (k-fold vs leave one cohort out vs past cohorts only)
    - Once we've gotten the cohorts we're using for model training and parameter estimation (demonstrated in green in the image above), we have a few choices on how to do parameter estimation.
    - K-Fold (5 fold). This is the standard technique. The risk with this is that it uses information from all cohorts to train on, which may mislead the performance estimates. For example, using information from other members of the 2009 cohort to predict on a student in the 2009 cohort is not a realistic scenario.
    - Leave One Cohort Out. To alleviate this, we could also ensure that cohorts remain separate. This is similar to K-Fold, where K is the number of cohorts and each fold is one cohort. This prevents using within cohort information to predict. However, it is susceptible to using information from future cohorts to predict on past students.
    - Past Cohorts Only. To address the issue of using future information, we can enforce that only past cohorts can be used for prediction and estimate the best parameter that way. For example, with 5 cohorts, this would involve 4 performance estimates for a given parameter value, with estimates using 1, 2, 3, and then 4 cohorts for training.
- Imputation technique for missing data in continuous features
    - Median value of the feature inserted for missing cells.

    - Mean value of the feature inserted for missing cells.
    - In a few cases where features are counts of rare events such as absences, missing values were set to zero in the feature generation process separately from this choice of hyperparameter.
- Scaling technique (from scikit-learn) used to compare features
    - Standard (mean-centered and scaled by estimated variance)
    - Robust (median-centered and scaled by interquartile range)
- Downsampling rate of the dominant class (on-time graduates)
    - None. All on-time graduates are kept in the training.
    - 0.90. A random 90% of on-time graduates are kept in the training.
    - 0.80. A random 80% of on-time graduates are kept in the training.
- Set of features to use
    - All features. Intuitively, this will result in our best performance. It's possible that there may be some noise around this, or that using all features could use to some misleading estimates though. To explore and understand our data better though, we explore our performance with different available features though.
    - Individual feature sets only
        - Absences only, grades only, snapshots only, mobility only, OAA only
    - Basics, non-detailed data
        - Total absences only
        - Overall GPA only
        - Snapshots
        - OAA normalized

Given some choices in all of these categories, we have a function to identify the best performing model family and the corresponding optimal tuning parameter. `*estimate_prediction_model.py*` takes the above meta-parameters in as a YAML options file, and then loops over the possible models and parameters.

- Model family examples: random forests, logistic regression, SVM, decision tree, gradient boosting, adaboosting, naive bayes
- Each model family has a corresponding grid of possible parameter values. The possible grid of parameter values to search over is noted in a separate `*grid_options.yaml*`.

With this understanding, we executed the following batches of models.

08_05_2016 / “Batch 1” (*initial exploration of meta-parameters*)

- Predicted only at beginning of 10th grade
- All possible ranges of student history from 9th grade only (5 training cohorts) to 5th - 9th grade (1 training cohort)
- All our possible model families, with only the default parameter
- Both possible imputation schemes
- Both possible scaling schemes
- All three cross-validation parameter estimation schemes
- Each individual feature set + All features
- No downsampling

08_09_2016 / “Batch 2” (*identifying how predictions change based on time of prediction*)

- Predicted at the beginning of each grade level, between 6th and 10th.
- All possible ranges of student history, with what's available depending on time of prediction.

- A smaller set of model families based on Batch 1 results: {logit, RF, SVM, GB, ET, DT}
    - Using `grid_options_small.yaml`.
- Median imputation
- Robust scaling
- All three cross-validation parameter estimation schemes
- All features
- No downsampling

NOTE: Batch 1 & 2 graphics and results are not quite accurate in the Repo, especially for recall under decision trees or naive bayes. This is because we previously used a default function to choose our "top-k-percent" of students. When there are ties, this function used to include all values, naturally increasing the recall level. Batch 2 also incorrectly includes OAA tests from future OAA scores (e.g. 8th grade) in an earlier model (e.g. 6th grade model).

08_12_2016 / "Batch 3" (*used to avoid an issue with batch 2 that incorrectly used future OAA scores and included the new `definite_plus_ogt` outcome*)

- Predicted at the beginning of each grade level, between 6th and 10th.
- Using only 1 or 2 years of student history at time of prediction
- A smaller set of model families based on Batch 1 results: {logit, RF, SVM, ET, DT}
    - Using `grid_options_small.yaml`
- Median imputation
- Robust scaling
- K-Fold cross validation
- All features + Basic non-detailed options
- All three downsample rates (.80, .90, and no downsampling)

08_15_2016 / "Batch 4" (*Focusing on LR and RF, without dropping reference category in dummy variables for logistic regression, and without dummifying categoricals in Random Forest*)

- Predicted at the beginning of each grade level, between 6th and 10th.
- Using only 1 or 2 years of student history at time of prediction
- A smaller set of model families based on Batch 3 results: {logit, RF}
    - Using `grid_options_small.yaml`
- Median imputation
- Robust scaling
- K-Fold cross validation
- All features
- Downsample rate of 0.80
- Reference category not dropped
- Random forest not dummified

08_17_2016 / "Batch 5" (*used to re-run the best models from batch 3, saving the order of the features in the pickle files for predictions on current unlabeled students)*

- Predicted at the beginning of each grade level, between 6th and 10th.
- Using only 1 year of student history at time of prediction, except for grade 7 which uses 2 years
- A smaller set of model families based on Batch 3 results: {logit, RF}
    - Using `grid_options_small.yaml`
- Median imputation
- Robust scaling

- K-Fold cross validation
- All features (except for grades 9 and 10, which have all features plus basic)
- Downsample rate of 0.80
- Reference category not dropped
- Random forest not dummified

In the following section, we discuss how we evaluated all of these various model runs as well as our insights and lessons learned through this process.

Here we list a handful of other model estimation setups that we would have liked to try:

- Hold the training cohorts fixed and vary the amount of student history. This should always show that more history improves performance. The interesting thing to study is to isolate the effect of how much history makes a difference by controlling for the amount of cohorts available. It's possible that the effect of including more or less student history changes depending on the time of prediction.
- Closer analysis of how use of different feature sets affects predictive performance. A clearer analysis of this can help MVESC make the case to its districts to collect more granular data. For example, using the IEP accommodation or teacher/classroom data (which we did not use), could be helpful in prediction and encourage more schools to provide that data. We also did not study the value of fine-grain absence data in detail, which is something MVESC was interested in at the beginning. Because many of these fine-grained datasets are available only for a subset of years and/or districts, this analysis would likely require comparing model performance trained on just these specific subsets.

IV. Evaluation Methodology

Analysis of Model "Meta"-Parameters

As described in the section above, there are several "meta"-parameters or options for training machine learning models we need to decide on. For every "meta"-parameter, except for the time of prediction, we use a data-driven approach, fitting different models on the training sets, and seeing which option leads to the best performance on the Validation Set. For the time of prediction, we want to build and keep a model for the beginning of each grade level. This will enable earlier predictions and allow us to explore what the tradeoffs look like when performing earlier prediction.

We chose median imputation and robust scaling because it showed similar or slightly better performance in the Batch One models we estimated. Since performance was mostly similar, we chose these because we feel they are more robust and safe than using mean-imputation or standard scaling, which could be susceptible to outliers. [Here is one image from the batch one analysis on imputation & scaling.]

For the set of features to use, we naturally chose to use all the features in our final models. In Batch One, we attempted different feature sets for insight into each feature set. Beyond getting a rough sense of their differing predictive value (when used individually) [see detailed image and summary image], we did not pursue this further. In other Batches, we used an 'All' and 'Basics' feature sets, but a brief, cursory look revealed that they have similar performance. This could be because we didn't fully utilize our detailed data, or . This was not analyzed thoroughly though, so conclusions cannot be drawn.

For the choice of cross-validation technique for parameter choice, we chose to use K-Fold cross validation. In Batch Two, K-Fold shows similar, if not slightly better performance, on the validation set. Also, K-Fold must be used when we only have one cohort of training data, so for consistency it makes sense to use across years. However, we recommend that the choice of cross-validation technique be checked again on the full data with our other options. We made the choice solely off of our Batch Two limited analysis, which was limited to only grades 7 and 8 (and prior to some code improvements).

For the grade range of data, from Batch One analysis, we chose to include only one or two years of historical student data at the time of prediction. The analysis of this batch showed that these two options usually resulted in the best performance on the Validation Set. There is going to be a natural tradeoff between years of historical student data and the number of cohorts available for training. These Batch Two results suggest that more cohorts are more helpful. Again, since this is an important option, we suggest future research to investigate this option again to ensure the robustness of this choice.

Finally, Batch Three was used to help choose the downsampling rate and to correct a bug allowing leakage of standardized OAA test scores from after the prediction point into the feature set. However, although these downsampling rates were recorded in the yaml and pickle files for Batch Three, they were not written to the database. In Batch Four, we chose an 80% downsampling rate because the best performing Batch Three models in cross-validation had an 80% downsampling rate (the negative class was downsampled so as to make up 80% of the total sample size. Of course, this choice should be re-evaluated in future research. 80% is also the lowest proportion we tested, so it may be worth investigating balancing the classes even more closely.

Using Avg Recall in Top 5-15% As The Evaluation Metric

So far, we have not clarified what we mean by “prediction performance”. Though we looked at full precision recall curves for our models, we chose the average recall when classifying the top 5 to 15% of students as ‘at-risk’ as our prediction evaluation metric. This follows two important assumptions.

First is the assumption that the intervention budget is fixed. We assume that schools can help some percentage of the highest estimated risk students. If we assume the budget is fixed and that we want to use the entire budget — since that intuitively leads to the biggest impact — then the “Recall at the Top K%” or “Precision at Top K%” metrics come to mind. This is a little unusual in machine learning, since we no longer need to consider the precision vs recall tradeoff; efficiency is already encoded into the recall at top k% metric by virtue of the fixed number of interventions provided.

In fact, if the K% to intervene on is a fixed number, then the relative ranking of model performance using Recall at Top K% or Precision at Top K% will be identical. This is because the denominators of both values are both fixed when K is fixed, while both have the identical numerator, True Detected Positives. Recall’s denominator is fixed by the Total Ground Truth Positives in the sample and Precision’s denominator is fixed by K, which is by definition equal to the number of Total Positive Detections. Thus, we can simply optimize for either recall or precision, and the model rankings are equivalent. This does not quite hold true for the average recall metric we settled on, as the equivalency of rankings only holds at a particular K, so for the average 5-15% metric we used recall to rank models.

The second assumption is that the fixed budget is unknown, but has uniform probability to be anywhere between 5 and 15% of students. This assumption is useful because we are building a model for several different school districts, each of which that can have different budgets for intervention. Our partner MVESC says to expect a range of intervention capacity across schools. Thus, we simply estimate the recall at 5%, 6%, …, 15% and take the unweighted average of these values; this is our evaluation metric of performance.

If, instead, we assumed that each school could choose a value of K between 5 and 15%, then our evaluation would be different. If K could be chosen, then a school might take into account efficiency and the tradeoff between precision and recall, such as putting more emphasis on higher precision by reducing

K% or higher recall by increasing K%. In such case, a different evaluation criteria is needed, perhaps one based on the ROC curve.

Identifying the Best Performing Model Families

As visualized for these various batches, we find Random Forest to generally be the best performing model family. We prefer to stick with one model family over all grade levels for consistency and understanding — it also streamlines the process of identifying feature importance. (However, it should be noted that this model selection decision was based on the performance on the original three choices of outcome targets: *definite*, *not on time*, and *is dropout*, rather than on the final *definite plus ogt* outcome used as a target in the best performing models selected for further evaluation and discussion. [Here is an example of model performance from Batch Three.]

Further, we also chose an L1 or L2 penalized logistic regression as our other best performing model family. It shows similar, or at times better, performance than the Random Forest models. Given this similar performance, we valued keeping the logistic regression model because of its familiarity and level of interpretability, especially with respect to student level feature importances.

Optimal Parameters

We performed some analysis of which parameters are optimal for these model families. However, this analysis is incomplete and we did not come to any solid conclusions. There is initial code for this parameter analysis and visualization here, which was performed just on the second batch.

Interpreting Individual Risk/Sensitivity Factors for Models

There are some useful lessons and ways to think about a model's individual risk factors. Most notably, there is an important and subtle difference between:

(A) risk factors, the features that contribute to a high individual estimated risk score, and

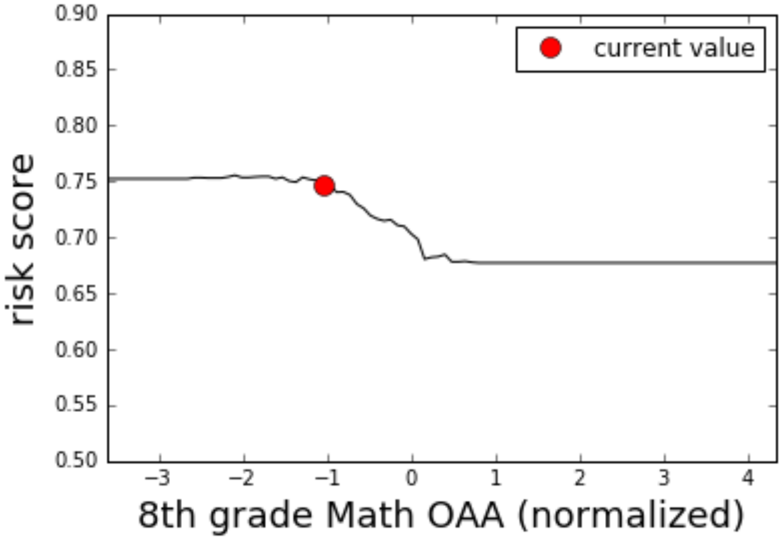
(B) risk score sensitivity, the features in the model that an individual risk score is sensitive to.

Both A and B could be interesting for teachers. A tells teachers what lead the model to predict a high risk score. B tells teachers in what features, for a specific student, the predicted risk is sensitive to small changes in. (Mathematically, B can also be thought of as an approximation of the partial derivative of the risk score.) Importantly though, it is important to interpret neither A nor B as **causal**. These are simply causal in how it affects our fitted models. It's important to continually recognize this since we have found the temptation to fall into a causal interpretation. In addition, all of these approaches fail to account for the effect highly correlated features, of which there are several among our features.

For a logistic regression, estimating both risk factors and risk sensitivity involve the estimated beta coefficients. A is the product of an individual feature value and the beta coefficients, while B is just represented by the beta coefficients. The highest features for A (risk factors) can differ amongst students, but for B (sensitivity), the highest beta coefficients will be the most sensitive. For example, perhaps the coefficient of the feature indicating number of discipline incidents is only moderately high, but if a particular student has an extremely high number of incidents that feature could be the most important factor in that particular risk score.

It may seem weird that B is the same for all students, but this is because of the strict additivity assumption of the logistic regression model. (Mathematically, this is because the partial derivative for a logistic regression usually is just the beta coefficient.)

For a random forest model, we cannot estimate A. The complexity and tree structure of the model make it hard to determine which particular feature resulted in a high individual risk score. We can however estimate B, risk score sensitivity in the random forest. We do this for each individual by taking their observed feature values, and creating several simulated permutations of that individual, changing one feature at a time by one small step size or (change in categorical feature). After seeing the resulting predictions from the fitted random forest, we can see which slight permutations resulted in the biggest changes in risk score. We can deem these as the highest risk sensitivities for the individual. This allows for some interpretability into the inner workings of the random forest.



Caption: This image shows the effect on one particular student's risk score when their 8th grade math OAA score is changed, all else held equal. The red dot show the actual value of this student's score, so we can see that increasing their score closer to the mean would reduce the risk score given by our model by a large amount.

We have focused on magnitude of change, but direction of change is important to consider. In the logistic regression example, do we only consider the highest positive product (risk factor)? The positive factors are the "risk" factors that influence the model to estimate an individual to be risky. On the other hand, the negative products are "protective" factors. A high negative value influences the logistic model to reduce a student's estimated risk. They aren't directly causing a student to be deemed risky, but if those features values are changed, it would suggest a large change in risk. Thus these strong "protective" factors may signal that these feature values are important to keep stable.

Similarly, for the random forest, since we can only calculate sensitivity (not separable risk factors), we consider the both highest magnitude "risk" and "protective" risk sensitivities. Respectively, they highlight the values where a slight change will notably decrease the estimated individual risk, and where a slight change will notably increase the estimated individual risk. On the output spreadsheet, a design choice should be made in how to present these.

Finally, comparing continuous and categorical features is also difficult. All the continuous features are compared based on fixed, small step sizes in the scaling standard deviation. However, there's no clear comparison between a small step size versus changing the categorical feature value. One option is to report the top few continuous sensitivity factors alongside the top few categorical sensitivity factors. This may make it easier than mixing the two types of features together. However, an interpretable point of comparison and example would be very useful.

We have focused on individual risk/sensitivity factors. This makes sense because we are considering individual interventions. Since different features have different observed feature values, they are in different parts of the parameter space with possibly different corresponding sensitivities.

If we are considering a general intervention to apply to a pool risky students, instead of a student-specific intervention, we must change our perspective though. We should consider which features are *generally* sensitive in the population or subpopulation of interest, by generalizing across individuals. To do so, we consider the average partial effect/derivative across the given pool. A simple way to do so is to average the partial risk sensitivities across the pool of students and identify which features have the generally highest average magnitude.

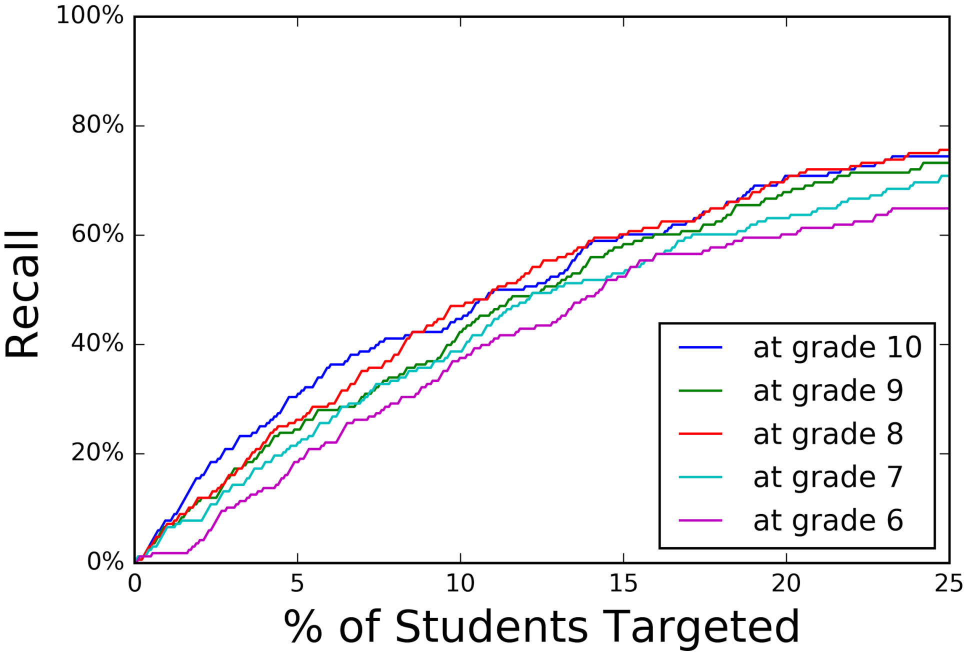
This could result in notably different takeaways. In a logistic regression example, Feature A could have the highest beta coefficient, but also only be relevant for students with very high risk scores, while Feature B could have a somewhat high coefficient and is relevant for students with a range of risk scores. On average, Feature B could have a higher average sensitivity though, since Feature A changes the log-odds for students with already very high log-odds, thus resulting in a relatively smaller change in the risk score. Generally, this principle could apply to any model family, where the ranking of individual feature importances may not be the same ranking across a more general subpopulation.

We have not analyzed average risk factors or sensitivities. As an extension, this could be useful if the schools are considering a more broad strokes intervention. Again, it's crucial to emphasize that these are not **causal** relationships, but simply the mechanics of the model.

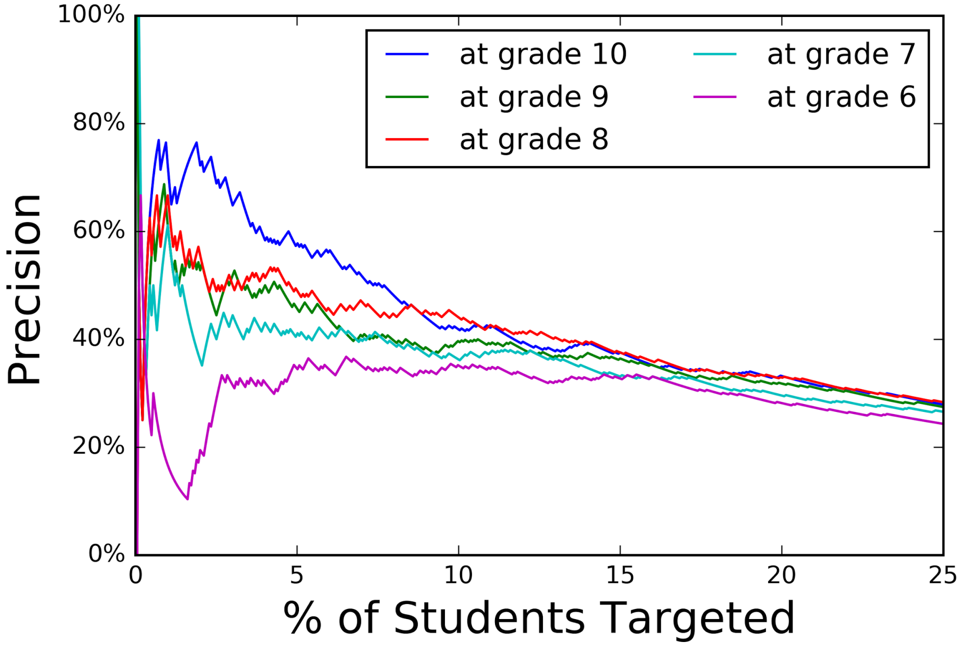
V. Results

Earlier Time of Prediction Is Viable

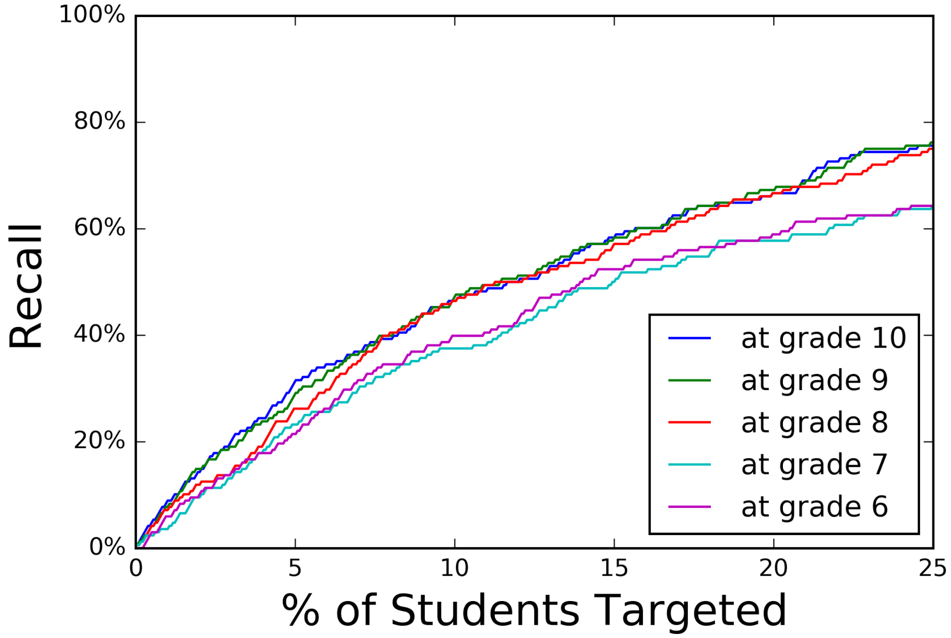
In comparing our best models for the beginning of each grade level, we find that there is not a large dropoff in performance when performing prediction earlier in a student's career. For example, comparing our best (beginning of) 10th grade model to our best (beginning of) 7th grade model, we can identify and intervene on 125 of 209 at-risk students at the start of 10th grade, but we can already identify and intervene on 111 of these students at the start of 7th grade. That is, of all the students that we are able to identify by 10th grade, we can find 89% of them three years earlier, affording the schools an additional three years for interventions to take effect. Comparing the 6th and 10th grade models, we find a similar trend. We can identify 97 out of 179 at-risk students by 10th grade, but of these, we can identify 92 of these students by the start of 6th grade. Although these comparisons are limited by the necessity of looking only at the intersection of students that appear on track within MVESC between 6th and 10th grade (excluding by necessity those that enter the district later, leave the district earlier, or do not advance with their cohort from 6th to 10th grades, and thus do not have predictions under both grade-level models), the overarching conclusion here is that, for the students we are able to identify at all, the vast majority of them (perhaps 89% or higher) could have been identified 3-4 years earlier.
We see similar results in the both the random forest models and the logistic regression models.



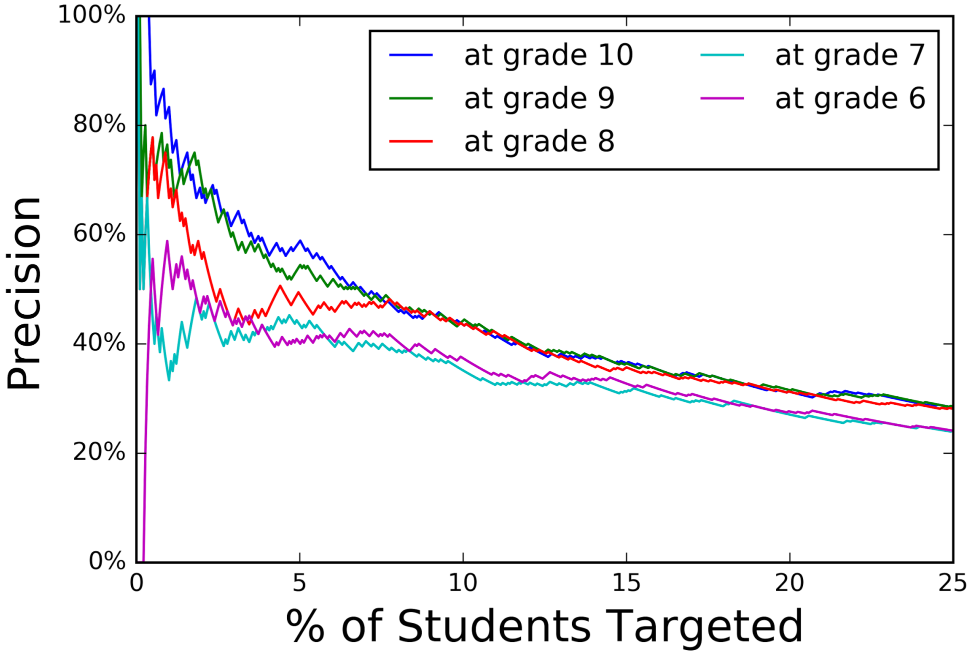
Random Forest model



Random Forest model



Logistic Regression model



Logistic Regression model

Caption:

Caption: In the 5-15% of the population range, our 6th grade model has an average recall of 36%. (399 out of 1107 total at-risk students over our sample)

In the 5-15% of the population range, our 10th grade model has an average recall of 42%. (465 out of 1107 total at-risk students over our sample)

The takeaway from this result is that for a large fraction of the at-risk students, we can provide them ~3 extra years of support. By identifying the at-risk students early, this gives teachers and schools these 3 extra years to help students get on-track, instead of waiting for the more extreme signals of being off-track that occur in high school. We anticipate that the increased intervention effectiveness that might stem from these additional years could more than compensate for the very slight reduction in recall. For the handful of students that we might miss (relative to our 10th grade model), we have the additional chance of identifying them at the start of 8th grade or 9th grade with the 8th and 9th grade models, providing additional opportunities for school districts to identify and provide support to students who may have been falling through the cracks of the additional support these districts hope to provide.

Using combined 7th and 8th grade models

In fact, some interesting preliminary results suggest that, while the 7th and 8th grade models individually have worse prediction performance than the 10th grade model (as expected), together, they might match or even outperform the 10th grade model. For those students who are present in the system for all of 7th, 8th, and 10th grades, the union of unique students identified as at-risk by the 7th grade model and the 8th grade model may successfully identify more at-risk students than those identified by the 10th grade model. If so, then more students can be helped with using a 7th and 8th grade model, two or three years earlier, instead of waiting for a more accurate 10th grade model. A caveat though: using a 7th and 8th grade model versus just a 10th grade model may imply intervening on a larger number of students, and therefore require a larger budget. [*Note: The reason this work is preliminary is that we used a 0.50 risk score cutoff to determine labels, but risk scores are not comparable across models. We are re-running it to use our usual 15% rank.*]

Future: Improving Comparison Across Grade Level Models

In the future steps, we discuss how we've compared the grade level models only on the intersection of students, which leaves only students who do not move in or out during the range.

Improvement on Baseline of Typical ABCs Approach

To put our model's performance in context, we worked with our partners to identify a good approximation of the rules of thumb being currently used. These are often referred to as the ABCs (attendance, behavior, and class grades). To construct a typical ABCs baseline as a comparison, we calculated percentiles separately for the subgroups of graduates and non-graduates and compared the number of absences, number of discipline incidents, and course grade point averages of students who graduate on time versus those who do not graduate. Based on these percentiles, we selected cutoffs that were exceeded by as many non-graduates as possible without incorrectly identifying too many graduates. We constructed this baseline model as a heuristic alternative to the 10th grade model, and so this baseline takes into account the student's aggregate attendance, discipline and grades over the course of 9th grade and generates a prediction at the start of 10th grade.The cutoffs determined by the data were:

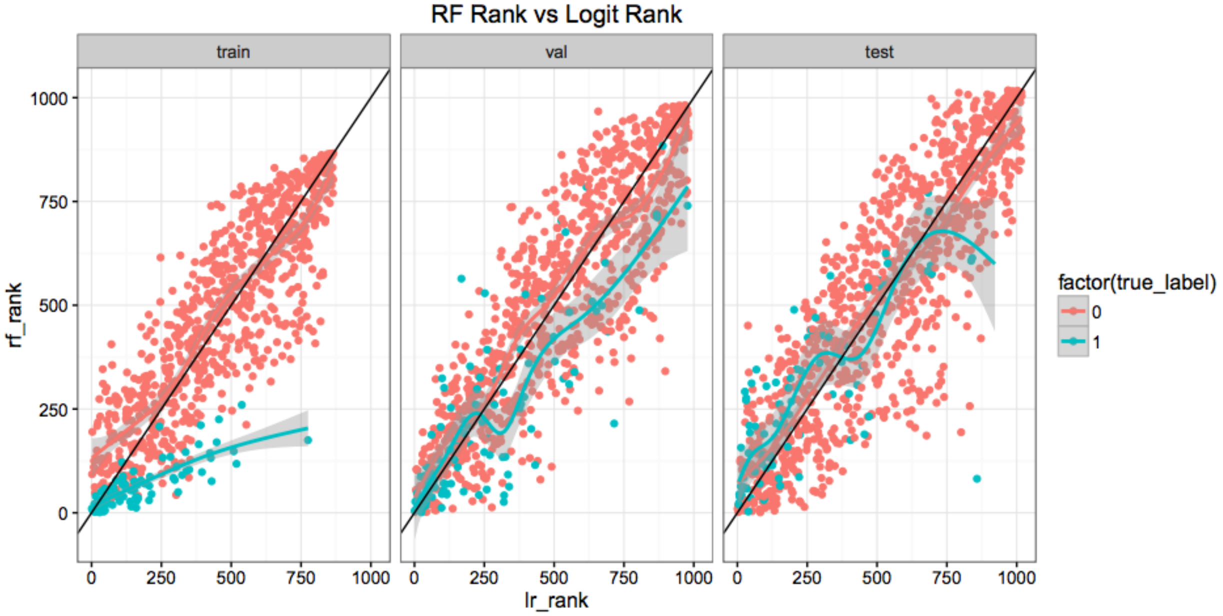
- Attendance: having 12 or more absences (of any kind) in 9th grade *[note that we used total absences rather than unexcused absences because unexcused absences were scarcely recorded and did not differentiate]*
- Discipline: having 2 or more discipline incidents (of any kind) in 9th grade
- Course Grades: Having a overall GPA in all subjects below 2.0

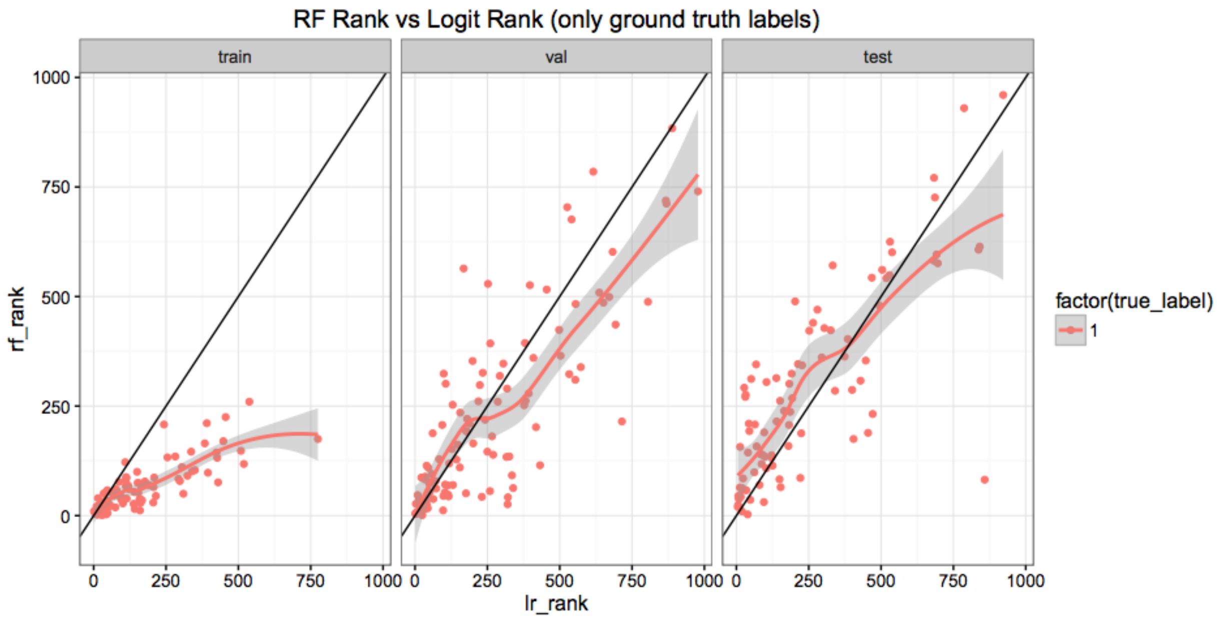
Using each cutoff individually, GPA by itself flags 58.7% of non-graduates compared to 14.8% of graduates. Discipline by itself flags 41.6% of non-graduates compared to 17.2% of graduates. Finally, attendance by itself flags 47.6% of non-graduates compared to 24.2% of graduates. However, combining these flags and identifying only students who exceed at least two or more of the three thresholds flags a total of 15% of the 10th grade student population, and of these flagged students, the baseline model achieves a precision of .325 and a recall of .437. For example, out of 2286 10th grade students, this system flags a total of 338 students, 110 of whom were actually at risk. By comparison, the random forest model on the same student population flagging the same number of students will correctly flag 148 of the 252 at-risk students, achieving a precision of .438 and a recall of .587. That is, in comparing the baseline model to the random forest model on all students entering 10th grade in 2013, our model correctly identifies an additional 15% of at-risk students that would not have been identified by the baseline heuristic flagging system.

Logistic Regression and Random Forest Can Differ Substantially

Comparing how the logistic regression and random forest rank students, there are a subset of students where the random forest and logistic regression can differ substantially. Student A could be ranked quite highly in risk by the random forest, but be in the middle of the pack of the logistic regression ranking; and vice versa. This suggests that it's possible that these two models are predicting well for different subgroups of students.

Here is an example of the rank difference in our final batch models for grade 6, with and without the negative label. As is clear below, the RF fits the training set quite well in ordering relative to the logistic, but this doesn't quite follow in the held out datasets. Further, there seems to be some large differences between the rank orderings of these two top models.:





A cursory investigation of the students where these models disagree most did not turn up obvious differences. A better approach that we would like to do is use another machine learning model on top to predict which students

Student Risk Scores Over Time

[Insert preliminary graphic of risk scores tracked by student over time.]

VI. Deployment Approach

The main deliverables are:

1) a spreadsheet of interpretable risk score predictions and models for students at different grade levels,
2) a recommendation for the doing earlier predictions based on prediction performance,
3) access to reproducible code for modeling pipeline and data management

Deliverable 1: Risk Scores Predictions

The sample deliverable can be seen at the above link.
For all the current students starting 6th through 10th grade in Fall 2016, we take our different models (one for each grade level) and estimate the risk for these current students. Using this, we create a spreadsheet for each school district, ranking the current students at-risk and not at-risk and classifying them as high risk, medium risk, low risk, and safe.

In the sample, each current student is a row in the spreadsheet. The sheet is sorted first by the school districts, school, and then by the students with the highest risk scores. For each student, we highlight the top 3 estimated risk factors for each student, and the corresponding value of that risk factor.

Deliverable 2: Tradeoff in Prediction Performance
Based on our results in the section above, we recommend using the predictive model as early as the beginning of 6th grade. We have provided and built models for 6th through the beginning of 10th grade. While prediction performance does increase as the time of prediction increases (logically), we believe it is useful to identify whichever students are identifiable as soon as possible. Then, going forward, any additional new students that are identified can be supported when they are identified. Of course, this is contingent on budget considerations too.

The source for this is based on our results above.

Deliverable 3: Reproducible Code and Pipeline
We prioritize re-usability. The pipeline has been engineered flexible for use in future years, alongside this technical report. The model should be able to be easily executed again after processing any new data snapshots. All of the code has been open sourced and documented on Github (https://www.github.com/dssg/mvesc).

Initial Field Testing

It's important that we take our predictive modeling outside of the "lab". With MVESC, we have discussed potential field testing schemes. Centrally, we want teachers to evaluate the risk of their current students and see how they compare with our model's estimates of risk.

Option one is to give teachers or school administrators a handout of our model predictions. MVESC can perform the task of translating the student lookup numbers to identifiable student names/IDs so that staff can connect each prediction to a specific student they know. In this option, we reveal our model's predictions, scores, and categorization of risk for each student. Then, we collect open feedback and notes on how teachers perceive the accuracy, insightfulness, and possible mistakes that the model is making. We can then use this to study and evaluate our models in a different light. Ultimately, since our end goal is to get these predictions into the hands and actual use of teachers, we want teachers and administrators to feel confident in these predictions. Their feedback will be helpful to shed light on potential flaws. However, we also want to be conscious of potential mistrust and biased feedback by teachers, for example, by revealing our predicted scores.

Option two is similar, except that we do not bias teachers by showing them our predicted scores. Instead, we can show them a handful of the data and high risk/sensitivity factors that we used, and allow them to come up with their own risk scores. Then, afterwards, we can compare our risk scores/rankings to theirs, and then see where there are meaningful or substantial differences.

MVESC Implementation Steps

Access to the database. We will provide a database dump of our cleaned and operational database, based on the original data provided.

Where will MVESC host the database?

Code. All of the code we used is stored on the Github repository. This is usable and downloadable. Depending on the amount of customization desired, it requires varying levels of knowledge of SQL and Python.

What process or venue does MVESC intend to use in order to use the output of the model to help in determining the student intervention/risk assessment plan?

08/23/2016 Notes:

Getting all the ~20% of at risk is totally fine. We'll do a proportional first pass for the 8,000 other safe students.

Any analysis on the impact of extracurriculars on student performance? Mike thinks this could be interesting.

Good to emphasize the strength factors, so it's interesting that we can show both risk and protective factors.

They want the error checking and failsafes in the code. Build some new code to check for the unique values that we've used for cleaning. In the report, clarify where it's important to check for errors and check for the data quality. "How do we know it's running really well". Document the suggested ideas for occasional checks when we get new data, or update the model, or re-train the model.

Strong voices in the ohio legislation attribute DoE to why the data collection is so messed up. Apparently, it's quite messy at the state level.

Highlight what MVESC can go ahead to run forward with this.

This is quite new for the area, so they want to continue to build and work on implementing this.

VI. Future Steps & Consideration for Improvement

- As MVESC looks to build on the initial work completed this summer, one theme that came up throughout the project is having MVESC focus on capturing additional data that can inform the model. There is a list in the appendix - **Recommended Datasets for MVESC to Capture in the Future** - that has a draft list of these items.

- In addition, MVESC may want to continue collaborating with the University of Chicago and other universities to improve upon the modeling. There is a list in the appendix - **Additional Tasks and Other Recommendations** - to document other ideas DSSG & MVESC came up with but did not have time to attempt in the summer.

Technical Lessons Learned Grab Bag

- Individual feature importance/risk factors are not equal to average feature importance/risk factors. Our choice of which to focus on depends on if we are making an individual-specific intervention, or if we are making a subgroup-specific intervention. The latter requires averaging over individual expected partial effects of changes / individual risk factors.

- Scaling functions in sci-kit-learn can make un-scaling difficult in Pandas, because it can permute the column ordering.
- Analyzing absences requires accounting for weekends appropriately
- Precision and Recall @ k% give equivalent orderings. They both have true_positives on the numerator, and the denominator is fixed for recall (all total positives) as well as for precision IF k is fixed (= the top k%). Thus, optimizing either is fine. Reporting can have some different connotations.
    - If we assume we have some flexibility in k though, then looking at the changes in precision/recall as we change k is valuable to consider efficiency.
      However, if budgets are fixed, then efficiency of intervention may not matter.
- MUST use intersection of students to compare precision and recall across grade levels. Because in each grade level, there are different underlying proportions of the outcomes.
- Tips for writing an effective and re-usable modeling pipeline.

Appendix

1. **Recommended Additional Data for MVESC to Capture**

- *Consistent, comprehensive labeling of outcomes* is crucial to an effective machine learning model. To this end, more complete documentation of transferring students would aid greatly in accurately identifying which students have dropped out. In particular, consistent withdrawal reason recording from all member school districts and incorporating statewide Ohio department of education data for better student outcome tracking could provide immense performance improvement.

2. **Additional Tasks and Other Recommendations**

- *Generate predictive model at mid-year, moving toward a continuously-updating model*
- Comprehensive data integrity tests (missing values tests, unique value tests) to automate checking compatibility of new data with the existing codebase
- Comparison of the risk scores generated by the DSSG models and the OSU models (i.e. rank correlation)
- Comparison of important features used in the DSSG models and the OSU models
- Use statewide data OSU has access to help clarify some of the outcomes for transfer students who leave the MVESC coverage area
- Analyzing the complementary nature of the datasets, for example the fine time-scale and overnight availability of some types of data in MVESC and the broad coverage area of the statewide data.
- Improving comparison amongst grade-level models. Currently, we have to focus on the intersection of students amongst the different grade-level models, which may unfairly focus just on students who do not move in or out of the area. We can identify some alternative comparisons, such as the recall at a fixed precision (or false positive rate).

- Focus on just a handful of cohorts to identify the effect of how historical data influences predictive performance.
- Focus on identifying the additive predictive performance of certain specific feature types - this should involve training/testing models using just the years/grades where the specific data is available (for example, the fine-grained absence data is only available in certain districts for specific years).
- Improve the spreadsheet output. For example, identify a better way to group students into categories. Within those categories, perhaps consider alphabetizing the students to avoid the rank order?
- Take a semi-supervised approach to labeling students who don't have clear outcomes in the data
- Geocode student and school addresses to incorporate ACS data
- Consider clustering or path analysis approaches to look at typical trajectories of students through their careers (some attempts at this for a previous, related project in the 2014 DSSG education project github repo)
- Analyze the differences between cohorts
- Explore the available teacher data to figure out what useful features might be derived. This data came too late in our process this summer for us to fully explore the possibilities.
- The random forest and logistic regression models have similar performance, but there are significant differences between the students they flag. Closer analysis of these differences, potentially using clustering or a decision tree model, could allow improvements in overall performance.
- A little more data cleaning could still happen, a few features such as section 504 plan and special ed indicators have a few unnecessary values, and some new school codes or student status indicators have not been mapped to appropriate interpretable values
- If we can get improved outcome data from clearinghouse or OSU or other sources, updating outcomes and analyzing differences in performance based on accuracy of labels
- Look at including fewer features by removing strongly correlated ones or doing some form of dimensionality reduction, but want to ensure that features remain easily interpretable
- More fine-grained error analysis: look at students who drop out but have low risk scores, consider how risk scores change over time
- Look at differences between districts, with respect to performance and data availability
- Generate and include other potential features listed above

3. Setup/Installation Documentation

  a. Schema of pipeline

  a. Database schema

  b. Walk through a setup with a DSaPP team member with new data file and new computer - does it run?

c. If your partner is going to be picking up your model/code/etc - what do they need to do to run it?

4. Other Links & Resources