Chicago Public Schools SQRP Playground

CAPP 30122, Winter 2020

Final Project

Alison Pelczar, Lily Grier, and Launa Greer

Overview

For our final project, we created a Django web application allowing users to experiment with the School Quality Rating Policy (SQRP) model used by Chicago Public Schools (CPS). The current SQRP system takes into account factors such as attendance, test score growth and attainment, and student and teacher survey responses on school climate; assigns each factor its own weight in determining an overall point score for a school; and then places the school in one of six tiers: “Level 1+,” “Level 1,” “Level 2+,” “Level 2,” “Level 3,” or “Inability to Rate.” This system has faced considerable criticism for incentivizing schools to push out students who could bring down ratings and penalizing schools that serve large numbers of students from underserved communities. The ability to quickly iterate through different versions of an SQRP model and analyze its effects on ratings would greatly benefit administrators as well as students and their families.

To address this need, we have created an interactive model of the SQRP that allows the user to select their own rating criteria from a menu of inputs of the current SQRP system, along with some additional inputs. Before launching the web app for the first time, a one-time setup process must be completed to populate two Sqlite3 database tables with school data. Next, the user selects and submits relative weights for each input on the first tab of the interface labeled “Configure Model.” Using these relative weights in conjunction with 2018-2019 school data stored in the database, the program calculates a rating for each school and reassigns weights for missing inputs as needed. It also calculates a bias score to communicate how well the schools’ ratings align with the demographics of their student bodies.

The interface receives these results from the backend and reports them in several different ways. A MapBox map shows the distribution of school levels geographically while the “View Results” tab displays a histogram of level counts and school rating records filterable by name, address, and level. The last tab, “View Bias Score,” displays and contextualizes the score and presents three regression plots showing the correlation between schools’ ratings and the percentage of English-language learner, low-income, and special education students. Ideally, schools’ ratings should not be easily predicted by greater concentrations of these groups.

Software Architecture

Our database, **db.sqlite3**, was created as part of the initial scaffolding for the Django web project. It was then populated through calls to two APIs: “Chicago Public Schools - School Progress Reports SY1819” and “Chicago Public Schools - School Profile Information SY1819,” hosted by the City of Chicago’s Open Data Portal. We also used data from the “SQRP Ratings and Accountability Status 2018-2019” Excel spreadsheet downloaded from the Chicago Public Schools website. Data from these three sources was put into Pandas dataframes and cleaned through the **core/clients/apiclient.py** and **core/clients/excel_client.py** files. The data was then uploaded to SQL database tables created and called by **core/clients/dbclient.py**. The dbclient.py file called the SQL setup script (**core/scripts/setup.sql**), which ensured the enrollment and sqrp tables were formatted properly. (Please see the appendix for more details on the data mapping.)

When the web application loads, the default results from the 2018-2019 school year are displayed as a starting point. Once a user selects which indicators to include and their relative weights, the input is passed through the web controller, **sqrp/views.py**. The controller parses the user input into an **SQRP** object and then sends the object to the calculate_sqrp_scores function in the **analyzesqrp.py**. This function pulls the school records and enrollment data from the database. Then, each of these records is used to create a **School** object, which includes location and contact info for the school and its points and rating under the user’s specified policy.

The points are calculated by the functions in **core/models/point_calculator.py**. If data for a particular indicator is missing, then the weight is reallocated according to the reassignment rules outlined on page 14 of the SQRP Handbook. These functions are in the **core/models/reassignments.py** file (see appendix). Depending on the combination of missing data and excluded indicators, it may not be possible to reallocate weight, in which case the numerical weights will not sum to 1. In this case, the numerical weight of each indicator is inflated proportional to its relative weight, and the point total is adjusted accordingly. The rating for the school is then assigned using the same cutoff points as in the current SQRP model. Schools which have a point value of 0 due to missing data are not given a rating, nor are four schools which serve exclusively special education students, which are not given a rating under the current model either.

The point totals are added to the enrollment dataframe. This dataframe is passed to the **core/models/bias_score.py** file, which calculates the bias score by running a linear regression with points as the dependent variable and percent of students who are low income, special education, and English language learners as the independent variables. Schools that did not receive a rating are excluded from the regression. The bias score is the R^2 value for this regression. More details about how we chose this method of calculating the bias score can be found in **core/docs/bias_score_dev.pdf**.

Finally, we generate two data visualizations. The first is a histogram showing the distribution of level ratings, and the second is a series of three regressions accompanying the bias score. Each regression is displayed as a scatterplot of an independent variable in the regression model against the point totals. These visualizations are saved in the project directory under the **sqrp/static/img** folder. When backend processing has completed, analyze_sqrp.py returns the list of school rating results and the bias score back to views.py. The controller then renders the home page whose routing is configured in **sqrp/urls.py**. The POST request contains a context object with the schools, bias score, bias score description, and current input form (configured in **sqrp/forms/forms.py**) passed in as fields. The fields are then parsed as template variables in the home page's HTML, **sqrp/templates/sqrp/home.html**. The page uses MapBox GL JS, the Google Materialize library, and a number of custom JavaScript functions to achieve the appearance and functionality of the user interface.

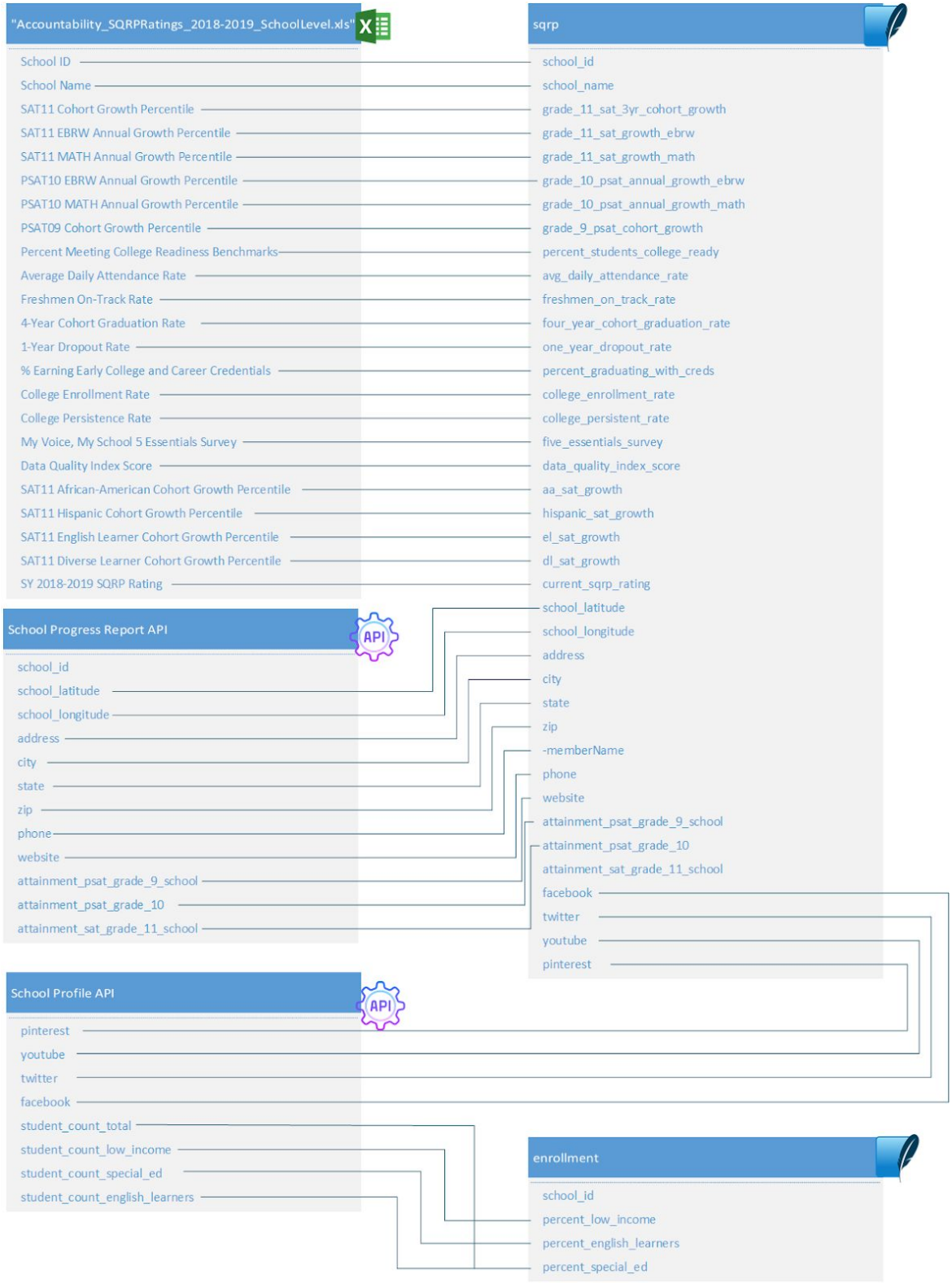
Accomplishments, Shortcomings, and Future Directions

Given time and resource constraints, there are many future directions for the project we were unable to pursue. Our goal was to create a tool for CPS administrators and stakeholders to evaluate the effects of different inputs on school rating outcomes, and we did accomplish this. If we had more time, we would add more indicators. We added standardized test attainment scores as one measure that is not in the current SQRP, but has been in past versions of the policy, and were able to see through our algorithm that this in fact led to increased bias. We would have liked to add more holistic indicators such as parental satisfaction, extracurricular offerings, and the extent to which a school facilitates meaningful interactions between students from different socioeconomic backgrounds. There is not sufficient data for many of these measures, so this would necessitate further data collection on the part of CPS.

On the user interface side, we would also like to enable filtering of results not just on school names, addresses, and levels, but on other factors, such as proximity to specified transit lines and whether the school accommodates certain learning needs. If another sprint of work were included in our project, we could also configure the map to hide and show markers whenever a user applied a new filter and troubleshoot slow page loading times.

Appendix

Data Architecture



SQRP Model Indicators

grade_11_sat_3yr_cohort_growth: the overall SAT growth rate for a school

grade_11_sat_growth_ebrw: the annual SAT EBRW (Evidence-Based Reading and Writing) section growth rate for the cohort of current 11th graders. The growth rate is calculated by comparing the average spring-to-spring scale score growth of 11th graders on the SAT EBRW section compared to the average national growth for schools with the same pretest score. The school is assigned a percentile representing where it would fall on the national distribution.

grade_11_sat_growth_math: the annual SAT math section growth rate for the cohort of current 11th graders. The growth rate is calculated by comparing the average spring-to-spring scale score growth of 11th graders on the SAT math section compared to the average national growth for schools with the same pretest score. The school is assigned a percentile representing where it would fall on the national distribution.

Grade_10_psat_annual_growth_ebrw: the annual PSAT EBRW (Evidence-Based Reading and Writing) section growth rate for the cohort of current 10th graders. The growth rate is calculated by comparing the average spring-to-spring scale score growth of 10th graders on the PSAT EBRW section compared to the average national growth for schools with the same pretest score. The school is assigned a percentile representing where it would fall on the national distribution.

grade_10_psat_annual_growth_math: the annual PSAT math section growth rate for the cohort of current 10th graders. The growth rate is calculated by comparing the average spring-to-spring scale score growth of 10th graders on the PSAT math section compared to the average national growth for schools with the same pretest score. The school is assigned a percentile representing where it would fall on the national distribution.

grade_9_psat_cohort_growth: the one-year PSAT growth rate for the cohort of current 9th graders. The growth rate is calculated from the spring PSAT composite scale score minus the average expected PSAT Composite scale score for 9th grade.

percent_students_college_ready: the percentage of students in the 9th, 10th, and 11th grades meeting or exceeding combined College Readiness Benchmarks established by the College Board.

avg_daily_attendance_rate: the average daily attendance rate of the school. The attendance rate is adjusted for students with qualifying medically fragile conditions, early graduation for 8th and 12th graders, transportation adjustments, and each school's two lowest attendance days only if the adjustment would improve the school's rate.

freshmen_on_track_rate: the percentage of students earning five or more credits and failing no more than 0.5 courses in a core subject during their 9th grade year.

four_year_cohort_graduation_rate: the percentage of students who were first-time freshmen four years prior who have graduated.

one_year_dropout_rate: the percentage of students in grades 9-12 dropping out during the year

percent_graduating_with_creds: the percentage of graduating students who have received early college or career credentials. Eligible credentials include: at least one credit from an approved early college course, a 3+ on an AP exam, a 4+ on an IB exam, the State Seal of Biliteracy, or an approved career certification

college_enrollment_rate: the percentage of students enrolled in a two- or four-year college in the fall or spring after graduation from high school

college_persistence_rate: the percentage of students enrolled in a two- or four-year college in the fall or spring after graduation from high school that remain enrolled in college the following fall

five_essentials_survey: the overall rating of the school on the "My School, My Voice 5 Essentials" survey administered in the spring to students and teachers. Schools must have a 50 percent response rate to receive a survey rating. The rating is determined using data from all five essentials, or from whatever combination of essentials for which the school has sufficient data.

data_quality_index_score: the percentage of data quality indicators that are correct in CPS data systems

priority_group_sat_growth: the relative weight for indicator 2 of 17: the SAT growth rate for students in the four priority groups. Priority groups include African American students, Hispanic students, English Learners and Diverse Learners. If there are fewer than 30 students in the priority group, the indicator is not used and the weight is reallocated to the SAT Cohort growth indicator.

attainment_psat_grade_9_school: the relative weight for the optional indicator grade 9 PSAT attainment. Not currently included in the CPS SQRP system.

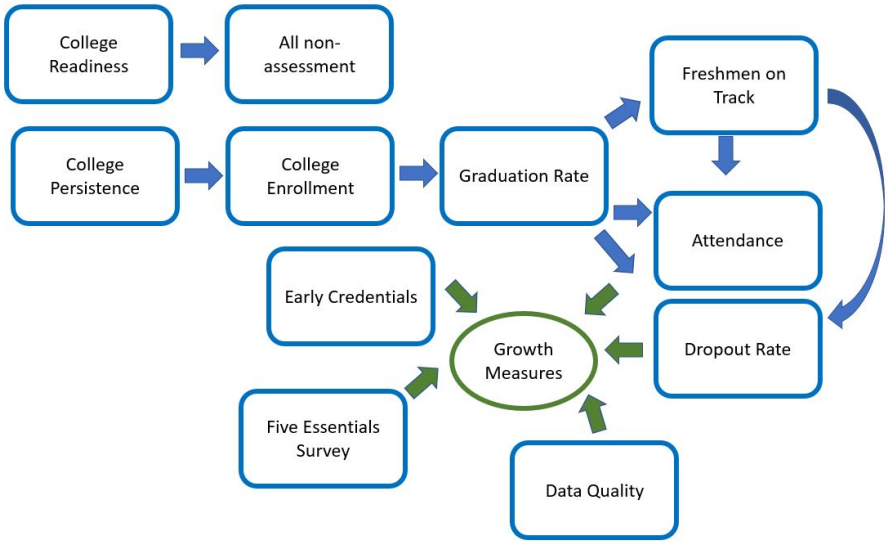
attainment_psat_grade_10: the relative weight for the optional indicator grade 10 PSAT attainment. Not currently included in the CPS SQRP system.

attainment_sat_grade_11_school: the relative weight for the optional indicator grade 11 SAT attainment. Not currently included in the CPS SQRP system.

Reassignment Logic

This flow is simplified, as weight is not always reassigned equally or proportionally.

Non-assessment indicators



Assessment and attainment indicators