This codebase consists of two folders: **degree completion** and **course completion**. Under each folder, the scripts can be executed in the following order to construct the study sample, build predictive models, and the perform bias analyses.

**Degree Completion**

1. **data\_construction subfolder**

**01\_identify\_truncated\_sample\_sizes.do**

This script finds out the distribution of currently enrolled students at VCCS in terms of enrollment lengths (how many terms have elapsed since the first term at VCCS).

**02\_identify\_truncation\_points\_through\_random\_sampling.ipynb**

This script performs random truncation to generate the observation window for each student, such that the distribution of enrollment lengths of our study is identical to the distribution of those for the currently enrolled cohort.

**03\_construct\_full\_dataset\_truncated.do**

This script processes all student level VCCS admin data to construct the data set which includes most of the predictors used for degree completion modeling.

**04\_create\_firstgen.do**

This script generates a special type of demographic predictors: The first-generation status: first-generation, non-first-generation, unknown.

**05\_create\_zip\_predictors.do**

This script creates a special type of demographic predictors: The predictors that are related to zip codes. They include student's distance to college, median households income associated with the zcta x year, percentage below poverty associated with the zcta x year. The missing value indicators showing the different types of the missing values are also generated.

**06\_find\_deglvl.do**

This script identifies the degree level each student is pursuing during each enrolled semester. This information will be used in analyses that have to do with degree-level-specific models.

**07\_find\_college\_program.do**

This script identifies the college and program (in terms of both CIP and CURR) for each observation, which will be used in both college-specific & program-specific models as well as exploratory analysis of the effects of college and program on demographic biases.

**08\_explore\_college\_program\_effects.ipynb**

This script performs various exploratory regression analyses to test the hypotheses as to whether the difference in enrollment patterns among different ethnic groups contributes to algorithm bias. In particular, this script also generates Appendix Table A8.

**09\_create\_original\_datasets.ipynb**

This script creates the full data set that includes all observations along with all predictor values (original + transformed/imputed) used in the RF model.

1. **Modeling\_and\_evaluation subfolder**

* Under “**base\_model**”:

**01\_fit\_base\_model.py**

This script fits the base random forest model for degree completion (on HPC) by identifying the opitmal set of hyperparameters.

**02\_evaluate\_base\_model.ipynb**

This script fits the optimized base RF model and then perform evaluation and bias analyses based on it.

**03\_mcFadden.ipynb**

This script computes the McFadden's adjusted R^2 for the training sample, based on the base model.

**04\_top10\_predictor\_comparison.ipynb**

This script computes the feature importance of predictors and perform comparison of base model's top 10 predictors in terms of Black vs. White groups.

**05\_base\_model\_predictions\_on\_specific\_test\_samples.ipynb**

This script uses the base model to generate the predicted scores on XX-specific (e.g. college-specific, first-term-specific) samples. Those XX-specific scores will be used in subsequent steps of comparing algorithmic biases.

**06\_generate\_calibration\_curves.ipynb**

This script generates the calibration curve plots (Figure 2), based on the base models.

**07\_quantify\_calibration\_bias.ipynb**

This script computes the quantified calibration bias at each threshold (in terms of predicted score percentiles) for the base model.

* Under “**gpa\_simple\_model**” and “**enrollment\_intensity\_simple\_model**”:

**01\_generate\_calibration\_curve.ipynb**

This script generates the calibration curve plots (Figure 4), based on the given simple models.

**02\_quantify\_calibration\_bias.ipynb**

This script computes the quantified calibration bias at each threshold (in terms of predicted score percentiles) for the given simple model.

* Under all other subfolders (for all other model variants):

**01\_fit\_model.py**

This script fits the given model variant for degree completion (on HPC) by identifying the optimal set of hyperparameters.

**02\_generate\_predictions.ipynb**

This script generates the predicted scores for the validation sample using the optimized parameters from the previous step.

**03\_quantify\_calibration\_bias.ipynb**

This script computes the quantified calibration bias at each threshold (in terms of predicted score percentiles) for the given model variant.

* Standalone script “**generate\_all\_predicted\_scores.ipynb**”:

This script generates predicted scores on the whole study sample (training + validation), based on the base RF model as well as all model variants.

**Course Completion**

1. **data\_construction subfolder**

**01\_create\_merged\_class\_data\_sample.do**

This script identifies all VCCS courses to be included in the study sample.

**02\_identify\_admin\_sample.do**

This script performs the initial step of identifying all observations from the VCCS admin data.

**03\_create\_section\_reated\_predictors.do**

This script generates the course section related predictors, such as section size and daytime/evening indicator.

**04\_preprocessing\_term-specific\_data.do**

This script takes in the raw student x term level VCCS data, performs necessary preprocessing steps to the data so that they can be later used to generate term-specific predictors.

**05\_find\_dual.do**

This script identifies the dual enrollment status for each student: Whether they have been dually enrolled or not.

**06\_find\_race.do**

This script generates the predictors which are binary indicators showing each student's gender and ethnicity.

**07\_identify\_instructors.do**

This script identifies the instructor associated with each course section.

**08\_create\_instructor\_related\_predictors.do**

This script generates the instructor-related predictors, such as full-time vs. part-time, as well as the average grade assigned by each instructor.

**09\_find\_prior\_terms\_gpa\_and\_enrl\_intensity.ipynb**

This script reorganizes the term-specific GPA and enrollment intensity data into dictionaries that map student x term to the lists of prior term GPA and term enrollment values, so that they can be processed later on to find out the trendline (slope) of term GPA and term enrollment intensity as predictors for grade prediction models.

**10\_create\_term\_gpa\_and\_enrl\_intensity\_predictors.do**

For each student x term, calculate the trendline (slope) predictors corresponding to prior term GPA and prior term enrollment intensity, using parallel processing.

**11\_trend\_predictors.ipynb**

This script aggregates all generated trendline predictors and convert them into the tabular format.

**12\_find\_college\_program.do**

This script identifies the college and program (in terms of CURR) for each observation, which will be used in both college-specific & program-specific models as well as exploratory analysis of the effects of college and program on demographic biases.

**13\_create\_firstgen.do**

This script generates a special type of demographic predictors: The first-generation status: first-generation, non-first-generation, unknown.

**14\_cip\_and\_degreetype.do**

This script identifies the program (in terms of CIP) as well as the degree type for each observation, which will be used in the exploratory analysis of the effects of program and degree types on demographic biases.

**15\_calc\_avg\_grades.do**

This script computes the average grade over the past five years, for each college x course x section.

**16\_parallelized\_prereq.py**

This script computes the average grade of the prerequisite course(s) of the target course, if applicable (using parallel processing).

**17\_parallelized\_prereq.py**

This script computes the average grade of the target course if the student has ever taken the course in the past, using parallel processing.

**18\_create\_pell\_by\_term.do**

This script identifies the pell-eligibility status for each student x term, which will be used in subsequent steps to generate pell-related predictors.

**19\_find\_pell\_status.do**

This script generates a special type of demographic predictors: If the student is pell-eligible in the most recent and penultimate terms, as well as if the student is ever pell-eligible.

**20\_clean\_zcta\_data.do**

This script performs necessary processing and cleaning processes for the ZCTA data.

**21\_create\_zip.do**

This script creates a special type of demographic predictors: The predictors that are related to zip codes. They include student's distance to college, median households income associated with the zcta x year, percentage below poverty associated with the zcta x year. The missing value indicators showing the different types of the missing values are also generated.

**22\_create\_demographics.do**

This script puts all generated demographic predictors together -- into a single table.

**23\_additional\_term\_specific\_predictors.do**

This script creates additional term-specific predictors, such as the share of credits taken online during the target term.

**24\_finalize\_term\_specific\_predictors.ipynb**

This script merges all generated term-specific together into one single table.

**25\_create\_cluster\_specific\_predictors.ipynb**

This script generates all cluster-specific (course-subject-specific) predictors.

**26\_create\_concurrent\_predictors.ipynb**

This script creates all of the predictors that are based on courses taken concurrently with the target course during the target term.

**27\_create\_course\_specific\_predictors.ipynb**

This script creates all of the course-specific predictors (with respect to the target course).

**28\_create\_training\_and\_validation\_sets.ipynb**

This script merges all types of predictors constructed together, and generates the finalized training and validation sets.

1. **Modeling\_and\_evaluation subfolder**

* Under “**base\_model**”:

**01\_fit\_model\_and\_generate\_predictions.py**

This script fits the base random forest model for course completion (on HPC) by identifying the opitmal set of hyperparameters, and then generates the predicted scores on the validation sample, and also performs basic evaluation of model performance and algorithmic biases (such as creating the calibration curve).

**02\_quantify\_calibration\_bias.ipynb**

This script computes the quantified calibration bias at each threshold (in terms of predicted score percentiles) for the base model.

**03\_identify\_feature\_importance.py**

This script outputs the feature importance for the base, base + racial predictors as well as race-specific models

**04\_top10\_predictor\_comparison.ipynb**

This script compares the base model's top10 predictors, in terms of Black vs. White groups.

**05\_mcFadden.ipynb**

This script computes the McFadden's adjusted R^2 for the training sample, based on the base model.

* Under “**gpa\_simple\_model**” and “**enrollment\_intensity\_simple\_model**”:

**01\_generate\_calibration\_curve.ipynb**

This script generates the calibration curve plots (Figure 4), based on the given simple models.

**02\_quantify\_calibration\_bias.ipynb**

This script computes the quantified calibration bias at each threshold (in terms of predicted score percentiles) for the given simple model.

* Under all other subfolders (for all other model variants):

01\_fit\_model\_and\_generate\_predictions.py

This script fits the given model variant for degree completion (on HPC) by identifying the optimal set of hyperparameters, and then generates the predicted scores for the validation sample using the optimized parameters.

02\_quantify\_calibration\_bias.ipynb

This script computes the quantified calibration bias at each threshold (in terms of predicted score percentiles) for the given model variant.

* Standalone script “**generate\_all\_predicted\_scores.ipynb**”:

This script generates predicted scores on the whole study sample (training + validation), based on the base RF model as well as all model variants.