VARIABILI:

y = CDR3 = Three-year cohort default rate 🡪 A 3-year cohort default rate is the percentage of a school's borrowers who enter repayment on certain Federal Family Education Loan (FFEL) Program or William D. Ford Federal Direct Loan (Direct Loan) Program loans during a particular federal fiscal year (FY), October 1 to September 30, and default or meet other specified conditions prior to the end of the second following fiscal year. Please refer to the Cohort Default Rate Guide for a more in-depth description of cohort default rates and how the rates are calculated.

INDEPENDENT VARIABLES:

INSTNM = Institution name

SAT\_AVG = Average SAT equivalent score of students admitted 🡪 Too many missing values

C150\_4 & C150\_L4 = The proportion of full-time, first-time, degree/certificate-seeking undergraduates who completed a degree or certificate at the institution within 150 percent of normal time, calculated from the IPEDS Graduation Rates component. Separate metrics are calculated for 4-year institutions and less-than-4-year institutions. This metric is calculated as the number of full-time, first-time, degree/certificate-seeking undergraduates who completed a degree or certificate divided by the number of full-time, first-time, degree/certificate-seeking undergraduates in the corresponding completion rate cohort (D150\_4, D150\_L4).

COMPL\_RPY\_3YR\_RT = Three-year repayment rate for completers

NONCOM\_RPY\_3YR\_RT = Three-year repayment rate for non-completers

GRAD\_DEBT\_MDN = The median debt for students who have completed

PCTFLOAN = The proportion of undergraduates who received a federal loan in the academic year, calculated from values from the IPEDS Student Financial Aid component. Proportions are expressed as decimals rounded to four decimal places, so, for example, 0.1234 equals 12.34 percent. This metric is not available prior to the 2009-10 academic year.

MD\_EARN\_WNE\_P8 = Median earnings of students working and not enrolled 8 years after entry

MEDIAN\_HH\_INC = Median household income

COUNT\_WNE\_INC1\_P10 = Number of students working and not enrolled 10 years after entry in the lowest income tercile $0-$30,000

COUNT\_WNE\_INC2\_P10 = Number of students working and not enrolled 10 years after entry in the middle income tercile $30,001-$75,000

COUNT\_WNE\_INC3\_P10 = Number of students working and not enrolled 10 years after entry in the highest income tercile $75,001+

ICLEVEL = Level of institution

UGDS = Enrollment of undergraduate certificate/degree-seeking students

CONTROL = Control of institution (public, private, …)

COSTT4\_A = Average cost of attendance (academic year institutions)

COSTT4\_P = Average cost of attendance (program-year institutions)

HIGHDEG = highest degree awarded (0-1-2-3-4)

Maybe we can add these two variables:

LOCALE = Locale of institution

Variables created because some columns are complements:

COST = COSTT4\_A + COSTT4\_P

C150 = C150\_4 + C150\_L4 🡪 Proportion of full-time, first-time, degree/certificate-seeking undergraduates who completed a degree or certificate at the institution within 150 percent of normal time, for both 4-year institutions and less-than-4-year institutions.

STEPS:

* Divide dataset in train e test (rows with null values for CDR3 in the test set)
* Delete rows with ‘PrivacySuppressed’ values since these missing values may be non-random and then convert these variables into numeric columns
* Scatter plots to see correlation between numeric columns and dependent variables and see outliers
* Create new columns for COST and C150
* Anova test for discrete variables to check if these variables are significant
* Correlation matrix to check correlation between all the continuous variables in our dataset:

1. High correlation between COMPL\_RPY\_3YR\_RT and NONCOM\_RPY\_3YR\_RT
2. High correlation between COUNT\_WNE\_INC1\_P10, COUNT\_WNE\_INC2\_P10 and COUNT\_WNE\_INC3\_P10

🡪 Maybe we can delete some of these columns

* One hot encoding for the 3 discrete columns (CONTROL, ICLEVEL and HIGHDEG)
* Delete rows with more than 5 missing values
* Delete SAT\_AVG column because it contains too many missing values
* Impute missing values with Iterative Imputer: multivariate imputer estimates each feature from all the others. It is a strategy for imputing missing values by modeling each feature with missing values as a function of other features in a round-robin fashion. This approach involves defining a model to predict each missing feature as a function of all other features and to repeat this process of estimating feature values multiple times. The repetition allows the refined estimated values for other features to be used as input in subsequent iterations of predicting missing values. It is a regression problem where missing values are predicted. Each feature is imputed sequentially, one after the other, allowing prior imputed values to be used as part of a model in predicting subsequent features. It is iterative because this process is repeated multiple times, allowing ever improved estimates of missing values to be calculated as missing values across all features are estimated.
* Check the distribution of the target variable: it does not resemble a normal distribution 🡪 while with the ‘sqrt’ transformation it resembles more a normal distribution. We can try our models with both initial target variable and the transformed one, making a copy of the data and applying the transformation.
* Since our variables have different scales, we can transform them with ‘min max scaler’: it scales each feature to a given range, in our case 0-1. We can try our models with both the initial features and the standardized ones.
* Divide the training set in train and evaluation sets in order to see and compare the performance of our models
* Having concluded the preprocessing of the data, the main part of the analysis starts here. This involved the application of different models, using all the various combinations of transformations of the input data and the target variable in each case, and the optimization of these models.

The optimization of the parameters will be carried out using GridSearchCV (or RandomizedSearchCV). This approach tests all possible combinations exhaustively using cross-validation, but this means that it requires a huge amount of computations. Because of this, especially given the number of potential parameters of the function, only a subset will be tested. We will try: RandomForestRegressor, …

* Having used RandomForestRegressor we can also check the importance of each feature (we can then try to apply RandomForestRegressor using only the most important features to see if it makes better predictions)