Miners Group

NU University

Abstract

This document describes the typical phases of a project, the tasks involved with each phase, and an explanation of the outputs of these tasks.

Predicting Repayment of Education Loans

NU Big Data & Data Science Diploma – Final Project

Contents

[1. Project Team 2](#_Toc32351683)

[2. Implementation Model & Methodology 2](#_Toc32351684)

[3. Business Understanding 3](#_Toc32351685)

[3.1 Describing Problem Area 3](#_Toc32351686)

[3.2 Defining Business Objectives 3](#_Toc32351687)

[3.3 Business Success Criteria 4](#_Toc32351688)

[3.4 Data Sources & Knowledge Stores 4](#_Toc32351689)

[3.5 Project Plan 4](#_Toc32351690)

[3.6 Assessing Tools & Techniques 4](#_Toc32351691)

[4. Data Understanding 5](#_Toc32351692)

[4.1 Describing Data 5](#_Toc32351693)

[4.2 Exploring Data 5](#_Toc32351694)

[4.3 Verifying Data Quality 11](#_Toc32351695)

[5. Data Preparation 11](#_Toc32351696)

[5.1 Selecting Data (Including or Excluding Data) 11](#_Toc32351697)

[5.2 Cleaning Data 20](#_Toc32351698)

[5.3 Constructing New Data 21](#_Toc32351699)

[6. Modeling 21](#_Toc32351700)

[6.1 Selecting Modeling Techniques 21](#_Toc32351701)

[6.2 Generating a Test Design 21](#_Toc32351702)

[6.3 Building the Models 21](#_Toc32351703)

[6.3.1 Parameter Settings 21](#_Toc32351704)

[6.3.2 Running the Models 21](#_Toc32351705)

[6.4 Assessing the Model 21](#_Toc32351706)

[7. Evaluation 23](#_Toc32351707)

[7.1 Evaluating the Results 23](#_Toc32351708)

[8. Deployment 23](#_Toc32351709)

[8.1 System Main Components 23](#_Toc32351710)

[8.2 System Workflow 24](#_Toc32351711)

[9. References 25](#_Toc32351712)

# Project Team

“Miners” Team:

1. Heba Saad
2. HebatAllah Zakaria
3. Usama Atteya
4. Yehia Atef Dorgham

# Implementation Model & Methodology

* This report presents our work on building a model that predicts the students’ loan repayment rates.
* The dependent variable is present in the dataset in a disaggregated form and hence it is to be decided which one of them is to be used. These are continuous variables that contain many unidentified values. Our model will predict those values with good accuracy.
* This document describes the entire process in the form of CRISP-DM model which represents the standard life cycle model for any data mining project.
* The life cycle model consists of six phases as shown in the following diagram, with arrows indicating the most important and frequent dependencies between phases.



# Business Understanding

## Describing Problem Area

* Our project’s idea about Educational Loans Repayment
* Most students during their college education incur a significant amount of debt
* In an effort to make educational investments less speculative, the US Department of Education has matched information from the student financial aid system with federal tax returns to create the College Scorecard dataset.

## Defining Business Objectives

* Our objective is to use the current institutes' dataset to predict students’ ability to repay their educational loans by exploring different institutional features.
* The proposed predictive model will aid the decision-makers of the US government to minimize the risk of bad debts.
* Also, it will put the power in the hands of students and families to compare colleges and see a better vision of how many graduates at a particular school are able to pay back their student loans.
* The ability to pay back student loans is generally a good indicator of how well colleges and universities are preparing their graduates for the job market.

## Business Success Criteria

* Minimizing the loss probability due to non-repaid loans for Loan granting organizations.

## Data Sources & Knowledge Stores

* We depend on a public dataset, US Department of Education (<https://collegescorecard.ed.gov/data/>)
* College Scorecard data are provided through federal reporting from institutions, data on federal financial aid, and tax information.
* These data provide insights into the performance of schools that receive federal financial aid dollars and the outcomes of the students of those schools.

## Project Plan

This is a high-level plan shows the project timeline across the main six phases

|  |  |
| --- | --- |
| Phase | Time |
| Business Understanding | 1 Week |
| Data Understanding | 2 Weeks |
| Data Preparation | 2 Weeks |
| Modeling | 2 Week |
| Evaluation | 1 Week |
| Deployment | 1.5 Week |
| Documentation | 1. Week |

## 3.6 Assessing Tools & Techniques

* The tools and platforms, used to implement our system were as following:
  + My SQL Database
  + Sqoop
  + Hadoop ( HDFS )
  + PySpark
  + Jupyter
  + Flask
  + Azkaban (Workflow Scheduler)
  + GitHub repository
* The techniques used to implement our system were as following:
  + Data Exploration & Feature understanding
    - Correlation
    - Statistical graphs
    - Null values detection
    - Outliers detection
  + Data Preparation & Preprocessing
    - Handling data encrypted or masked due to privacy issue
    - Handling Null/Nan values detection (Imputation)
    - Categorical variables encoding
  + Models
    - XGBoost
    - Ridge
    - Lasso
    - Multiple Linear Regression
    - SVR
  + Model Evaluation & Assessment
    - R Square
    - RMSE
    - Cross-Validation (K-fold)

# Data Understanding

## Describing Data

* After review the data definition guide for this dataset, we found that data can be categorized into eight main categories as following:
  1. Basic information about the dataset (OPEID, Currently Operating…)
  2. About the School data(identifiers, location, degree type and profile, programs offered, and the academic profile of students enrolled)
  3. Academics and Admissions data
  4. Costs data (to evaluate the tradeoffs of access, affordability, and outcomes)
  5. Student data (family income, race, Part/full-time status…)
  6. Financial Aid data (including Pell Grants and federal student loans)
  7. Completion data (College completion is associated with other positive outcomes, like finding a job and successfully repaying student loans)
  8. Earnings and Repayment data

## Exploring Data

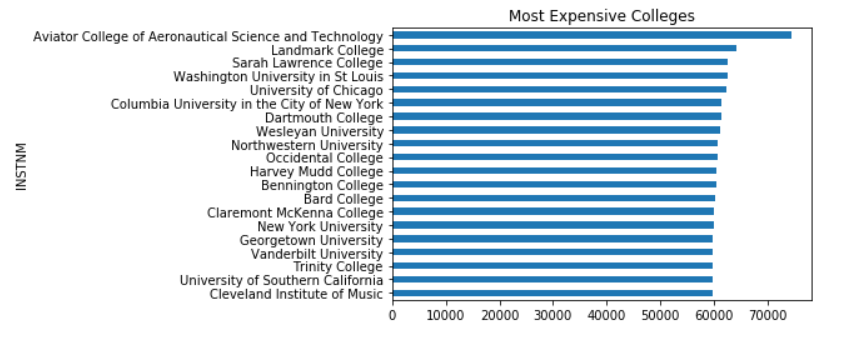
* Actually this data partitioned as 22 CSV file with **1,731** associated columns (variables/features), So we start to study
* There’s only one table in the dataset [Scorecard], It has over a hundred thousand rows in it **124699**.
* Each row corresponds to data on a US school in a given year. The years covered in this data range from 1996 to 2013
* This data is very wide and rich: it has 1731 associated columns (variables/features). One of the most interesting aspects of this data is the earnings information it contains on students once they graduate.
* We start to review data definition guide document in depth to:
  + - Understand these 1731 associated features, their type, their business meaning, and the relations between them.
    - Validate the data availability and consistency over the different years, as some features defined at specific years and obsolete at another one due to the changes in laws and business regulations.
* Then as per our initial understanding of the use case and data availability, we choose to:
  + - Start with a sample of data (e.g. Data of 2013-2014 academic year).
    - Select 56 features out of these 1731 associated features, those are candidates to affect the target of this use case.
* only suspected features (inputs and outputs) will be loaded to examine by graphs and statistics

|  |  |
| --- | --- |
| Feature | Description |
| INSTNM | The institution’s name |
| ADM\_RATE , ADM\_RATE\_ALL | **Admission Rate** For institutions with multiple branches, ADM\_RATE includes the admissions rate at each campus, while ADM\_RATE\_ALL represents the admissions rate across all campuses, defined as the total number of admitted undergraduates across all branches divided by the total number of undergraduates who applied across all branches |
| ACTCMMID ,  ACTENMID ,  ACTMTMID ,  ACTWRMID | ACT (ACT\*MID for CM, EN, MT, and WR) scores |
| SAT\_AVG ,  SAT\_AVG\_ALL | Average SAT scores for reading , writing and math |
| SATMTMID | Midian SAT Math score |
| UGDS | **Number of Undergraduate Students** Includes the number of degree/certificate-seeking undergraduates enrolled in the fall |
| HIGHDEG | **Degree Type** The highest award level conferred at the institution |
| CONTROL | Identifies whether the institution’s governance structure is public, private nonprofit, or private for-profit |
| INEXPFTE | Instructional expenditures per FTE student |
| AVGFACSAL | The average faculty salary |
| COSTT4\_P ,  COSTT4\_A | **Average Cost of Attendance, Tuition and Fees** The average annual cost of attendance includes tuition and fees, books and supplies, and living expenses for all full-time, first-time, degree-/certificate-seeking undergraduates who receive Title IV aid.  For academic year institutions (COSTT4\_A) and for program-year institutions (COSTT4\_P) |
| PCTFLOAN | **Percent of Undergraduates Receiving Federal Loans** Shows the share of undergraduate students who received federal loans in a given year. It can provide important context to figures related to debt, repayment, and non-repayment. This figure may be influenced by the eligibility for federal loans and the extent to which students apply for federal loans, as well as by the cost of the programs |
| PCTPELL | **Percentage of Pell Students** Shows the share of undergraduate students who received Pell Grants in a given year. This is an important measure of the access an institution provides to low-income students. However, it may not capture all low-income students |
| MEDIAN\_HH\_INC | Median household income |
| UGDS\_WHITE ,  UGDS\_BLACK ,  UGDS\_HISP ,  UGDS\_ASIAN ,  UGDS\_AIAN ,  UGDS\_NHPI ,  UGDS\_2MOR ,  UGDS\_NRA ,  UGDS\_UNKN | **Undergraduate Student Body by Race and Gender** This includes the total enrollment of undergraduate, degree-seeking students, based on fall enrollment, who are: men (UGDS\_MEN), women (UGDS\_WOMEN), white (UGDS\_WHITE), black (UGDS\_BLACK), Hispanic (UGDS\_HISP), Asian (UGDS\_ASIAN), American Indian/Alaska Native (UGDS\_AIAN), Native Hawaiian/Pacific Islander (UGDS\_NHPI), two or more races (UGDS\_2MOR), non-resident aliens (UGDS\_NRA), and race unknown (UGDS\_UNKN). |
| PPTUG\_EF | **Undergraduate Students by Part-Time/Full-Time Status** Includes the proportion of degree/certificate-seeking undergraduates enrolled part time in the fall term |
| TUITIONFEE\_IN ,  TUITIONFEE\_OUT ,  TUITIONFEE\_PROG | The cost data include the tuition and required fees of the institution. They are provided for in-state students (TUITIONFEE\_IN), out-of-state students (TUITIONFEE\_OUT), and program-year institutions (TUITIONFEE\_PROG). |
| TUITFTE | The net tuition revenue per full-time equivalent (FTE) student |
| DEATH\_YR3\_RT ,  DEATH\_YR4\_RT ,  COMP\_ORIG\_YR2\_RT ,  COMP\_ORIG\_YR3\_RT ,  COMP\_ORIG\_YR4\_RT ,  LOAN\_DEATH\_YR3\_RT ,  LOAN\_COMP\_ORIG\_YR3\_RT | **Completion and Transfer Rates** Each institution has all possible outcomes reported: share of students who died (DEATH\_YR\*\_RT), completed at the original institution (COMP\_ORIG\_YR\*\_RT), students who ever received a federal loan at the measured institution (LOAN\_\*) |
| AGE\_ENTRY | Entry age for students |
| COUNT\_NWNE\_P10 ,  COUNT\_WNE\_P10 |  |
| MN\_EARN\_WNE\_P10 ,  MD\_EARN\_WNE\_P10 | **Mean and Median Earnings** Mean (MN\_EARN\_WNE\_P\*) and median (MD\_EARN\_WNE\_P\*) earnings are for the institutional aggregate of all federally aided students who enroll in an institution 10 years after the student enrolls |
| COMPL\_RPY\_1YR\_RT , COMPL\_RPY\_3YR\_RT ,  COMPL\_RPY\_5YR\_RT ,  COMPL\_RPY\_7YR\_RT ,  NONCOM\_RPY\_1YR\_RT ,  NONCOM\_RPY\_3YR\_RT ,  NONCOM\_RPY\_5YR\_RT ,  NONCOM\_RPY\_7YR\_RT | **Repayment Rate on Federal Student Loans** These data are available for all borrowers at the institution, as well as disaggregated by completion status (COMPL\_RPY\_\* for students who completed and NONCOM\_RPY\_\* for students who withdrew without completing) |

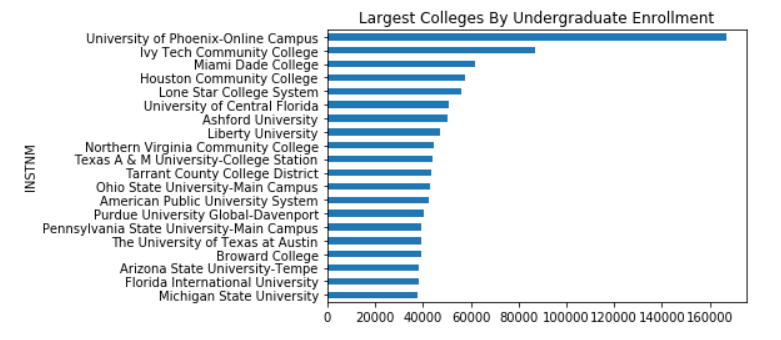
* Explore info of loaded data as following

|  |
| --- |
| Index: 7804 entries, Alabama A & M University to Georgia Military College-Stone Mountain  Data columns (total 53 columns):  dtypes: float64 (31), int64 (2), object (20) |

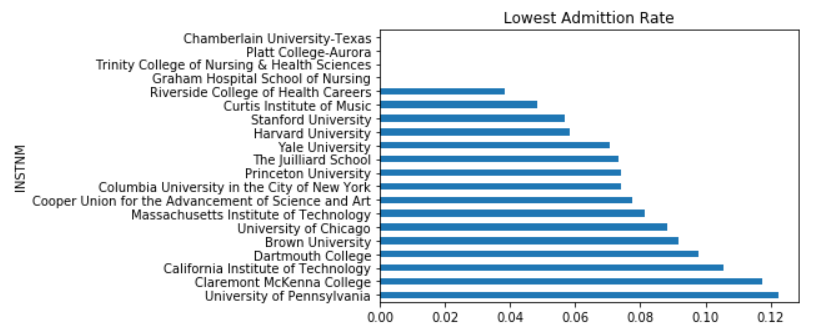
* Produce some graphs show some facts about US colleges.
  + Most expensive colleges
    - The cost here reflects the average annual total cost of attendance, including tuition and fees, books and supplies, and living expenses for all full-time, first-time, degree/certificate-seeking undergraduates who receive Title IV aid.



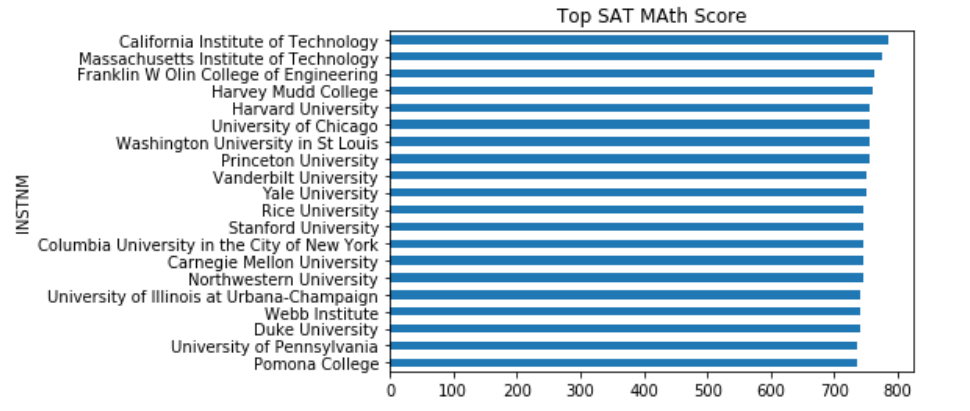
* + Colleges have the highest enrollment rate



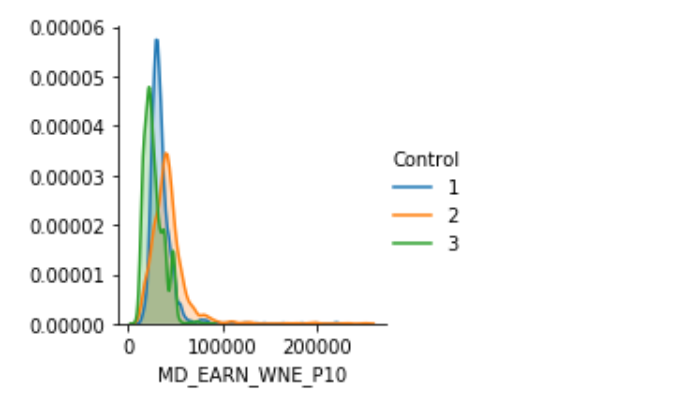
* + Colleges have the lowest admission rate
    - One of the most obvious measures of a school’s competitiveness is its admission rate.



* + Colleges have the top median SAT Math scores



* + Median earnings 10 years after matriculation as per college type / control
    - We can see the distribution of earnings 10 years after matriculation at undergraduate institutions. We’ll split this into three types of schools:
      * private non-profit [3],
      * private for-profit [2],
      * And public [1].
    - The median earnings for public and private non-profit schools look similar over the bulk of the earnings range.
    - At the very low end, there are more private non-profit schools than public schools. Curiously, the median earnings distribution for private for-profit schools is bimodal.



## Verifying Data Quality

* Since our problem is a regression problem, almost all suspected input features are float columns. But some columns have a value of "Privacy Suppressed" as an indicator that the value is missing for privacy reasons.

We found that these values must be replaced by null first to be able to convert these columns to float columns to run correlation, plotting functions safely.

|  |
| --- |
| def cleanPrivacySuppressed (dataFrame) |

* Also, we calculated the percentage of missing or null values across all candidate features to be able to decide an appropriate way to handle them. For all model except xgboost we replaced null values by median/mean of column.

# Data Preparation

## Selecting Data (Including or Excluding Data)

* As mentioned in section 4.2, we choose to start with 56 features out of these 1731 associated features these 56 features were selected by intuition according to our initial understanding of the available data and the business objectives.
* We tried to choose the best input features and target variable based on **Correlation and business use case**, So we generate the absolute correlation for all variables loaded in data-frame
* There is more than one output variable can be used as the target variable, suspected outputs are :
  + - COMPL\_RPY\_1YR\_RT (One-year repayment rate for completers)
    - COMPL\_RPY\_3YR\_RT (Three-year repayment rate for completers)
    - COMPL\_RPY\_7YR\_RT (Seven-year repayment rate for completers)
    - COMPL\_RPY\_5YR\_RT (Five-year repayment rate for completers)
    - NONCOM\_RPY\_3YR\_RT (Three-year repayment rate for non-completers)
    - NONCOM\_RPY\_5YR\_RT (Five-year repayment rate for non-completers)
    - NONCOM\_RPY\_1YR\_RT (One-year repayment rate for non-completers)
    - NONCOM\_RPY\_7YR\_RT (Seven-year repayment rate for non-completers)
* The output variable that has the highest correlation with the other input features will be chosen as the target variable, meanwhile, other candidates will not be considered as input features.

Assuming that the accepted correlation value between the output variable an any of input variables must be greater than **0.5**

* + For [COMPL\_RPY\_1YR\_RT]

|  |
| --- |
| COMPL\_RPY\_3YR\_RT 0.946264  NONCOM\_RPY\_3YR\_RT 0.927266  NONCOM\_RPY\_5YR\_RT 0.913966  NONCOM\_RPY\_1YR\_RT 0.906497  COMPL\_RPY\_5YR\_RT 0.901021  NONCOM\_RPY\_7YR\_RT 0.840605  COMPL\_RPY\_7YR\_RT 0.749900  PCTPELL 0.697547  SAT\_AVG\_ALL 0.668902  SAT\_AVG 0.651445  ACTCMMID 0.636761  ACTMTMID 0.628290  SATMTMID 0.626058  AVGFACSAL 0.607634  ACTENMID 0.605319  MD\_EARN\_WNE\_P10 0.591054  MN\_EARN\_WNE\_P10 0.590682  AGE\_ENTRY 0.581915  CONTROL 0.547110  LOAN\_DEATH\_YR3\_RT 0.536322 |

* + For [COMPL\_RPY\_3YR\_RT]

|  |
| --- |
| COMPL\_RPY\_5YR\_RT 0.952187  COMPL\_RPY\_1YR\_RT 0.946264  NONCOM\_RPY\_5YR\_RT 0.928185  NONCOM\_RPY\_3YR\_RT 0.907603  NONCOM\_RPY\_7YR\_RT 0.898931  NONCOM\_RPY\_1YR\_RT 0.865118  COMPL\_RPY\_7YR\_RT 0.851590  LOAN\_DEATH\_YR3\_RT 0.707518  PCTPELL 0.704221  MD\_EARN\_WNE\_P10 0.645223  SAT\_AVG\_ALL 0.641538  MN\_EARN\_WNE\_P10 0.640586  SAT\_AVG 0.623695  AVGFACSAL 0.615004  ACTCMMID 0.614424  ACTMTMID 0.606382  SATMTMID 0.591391  CONTROL 0.585123  ACTENMID 0.582368  AGE\_ENTRY 0.550468  HIGHDEG 0.535670  COMP\_ORIG\_YR2\_RT 0.507860 |

* + For [COMPL\_RPY\_5YR\_RT]

|  |
| --- |
| COMPL\_RPY\_3YR\_RT 0.952187  NONCOM\_RPY\_7YR\_RT 0.916724  NONCOM\_RPY\_5YR\_RT 0.908556  COMPL\_RPY\_1YR\_RT 0.901021  COMPL\_RPY\_7YR\_RT 0.890355  NONCOM\_RPY\_3YR\_RT 0.871874  NONCOM\_RPY\_1YR\_RT 0.820319  PCTPELL 0.686690  LOAN\_DEATH\_YR3\_RT 0.659897  MD\_EARN\_WNE\_P10 0.645418  MN\_EARN\_WNE\_P10 0.640329  SAT\_AVG\_ALL 0.625157  AVGFACSAL 0.624277  SAT\_AVG 0.618989  ACTCMMID 0.613180  ACTMTMID 0.605468  SATMTMID 0.593424  ACTENMID 0.581160  CONTROL 0.565434  AGE\_ENTRY 0.554506  COMP\_ORIG\_YR2\_RT 0.525298  HIGHDEG 0.518821 |

* + For [COMPL\_RPY\_7YR\_RT]

|  |
| --- |
| COMPL\_RPY\_5YR\_RT 0.890355  NONCOM\_RPY\_7YR\_RT 0.873590  COMPL\_RPY\_3YR\_RT 0.851590  NONCOM\_RPY\_5YR\_RT 0.803168  COMPL\_RPY\_1YR\_RT 0.749900  NONCOM\_RPY\_3YR\_RT 0.746289  LOAN\_DEATH\_YR3\_RT 0.712669  MD\_EARN\_WNE\_P10 0.707403  MN\_EARN\_WNE\_P10 0.696691  NONCOM\_RPY\_1YR\_RT 0.667975  COMP\_ORIG\_YR2\_RT 0.634738  PCTPELL 0.578859  AVGFACSAL 0.571819  SAT\_AVG\_ALL 0.566967  HIGHDEG 0.565509  ACTMTMID 0.548090  ACTCMMID 0.544406  SAT\_AVG 0.543223  SATMTMID 0.521642  ACTENMID 0.500940 |

* + For [NONCOM\_RPY\_1YR\_RT]

|  |
| --- |
| NONCOM\_RPY\_3YR\_RT 0.944075  COMPL\_RPY\_1YR\_RT 0.906497  NONCOM\_RPY\_5YR\_RT 0.904081  COMPL\_RPY\_3YR\_RT 0.865118  COMPL\_RPY\_5YR\_RT 0.820319  NONCOM\_RPY\_7YR\_RT 0.812678  PCTPELL 0.673040  COMPL\_RPY\_7YR\_RT 0.667975  SAT\_AVG 0.662759  SAT\_AVG\_ALL 0.653900  ACTCMMID 0.646307  ACTMTMID 0.636790  SATMTMID 0.635962  ACTENMID 0.621458  AVGFACSAL 0.588600  MN\_EARN\_WNE\_P10 0.586222  MD\_EARN\_WNE\_P10 0.584202  TUITIONFEE\_OUT 0.571291  AGE\_ENTRY 0.557894 |

* + For [NONCOM\_RPY\_3YR\_RT]

|  |
| --- |
| NONCOM\_RPY\_5YR\_RT 0.946722  NONCOM\_RPY\_1YR\_RT 0.944075  COMPL\_RPY\_1YR\_RT 0.927266  COMPL\_RPY\_3YR\_RT 0.907603  COMPL\_RPY\_5YR\_RT 0.871874  NONCOM\_RPY\_7YR\_RT 0.869925  COMPL\_RPY\_7YR\_RT 0.746289  PCTPELL 0.711239  SAT\_AVG 0.653461  SAT\_AVG\_ALL 0.647873  ACTCMMID 0.646958  ACTMTMID 0.644148  MD\_EARN\_WNE\_P10 0.629235  AVGFACSAL 0.628608  MN\_EARN\_WNE\_P10 0.625422  ACTENMID 0.623343  SATMTMID 0.620496  TUITIONFEE\_OUT 0.569642  AGE\_ENTRY 0.562711  CONTROL 0.533233  HIGHDEG 0.509697 |

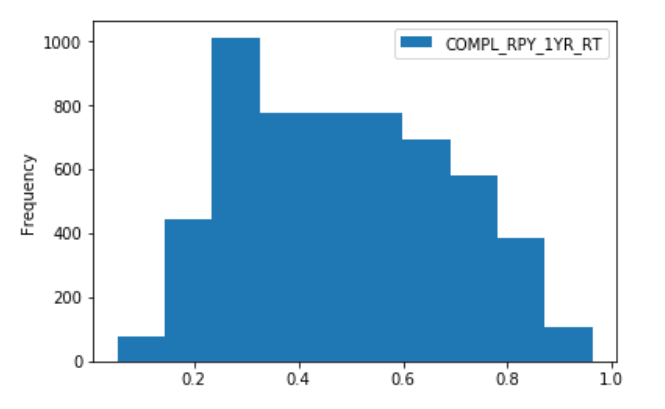
* + For [NONCOM\_RPY\_5YR\_RT]

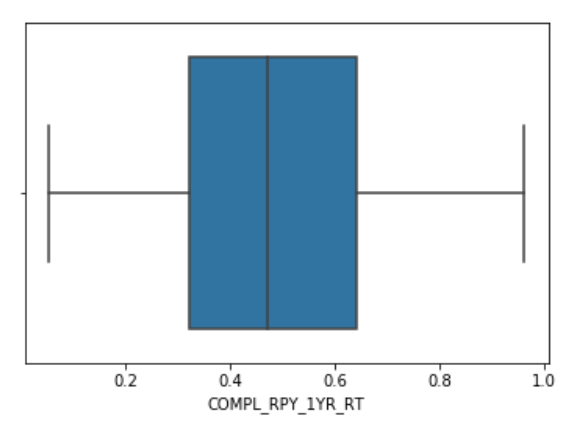
|  |
| --- |
| NONCOM\_RPY\_3YR\_RT 0.946722  COMPL\_RPY\_3YR\_RT 0.928185  NONCOM\_RPY\_7YR\_RT 0.923287  COMPL\_RPY\_1YR\_RT 0.913966  COMPL\_RPY\_5YR\_RT 0.908556  NONCOM\_RPY\_1YR\_RT 0.904081  COMPL\_RPY\_7YR\_RT 0.803168  PCTPELL 0.722585  AVGFACSAL 0.652072  SAT\_AVG\_ALL 0.650532  MD\_EARN\_WNE\_P10 0.650422  SAT\_AVG 0.648010  MN\_EARN\_WNE\_P10 0.645042  ACTCMMID 0.638323  ACTMTMID 0.633066  SATMTMID 0.618226  ACTENMID 0.604815  AGE\_ENTRY 0.571624  CONTROL 0.557939  TUITIONFEE\_OUT 0.550776  HIGHDEG 0.529304 |

* + For [NONCOM\_RPY\_7YR\_RT]

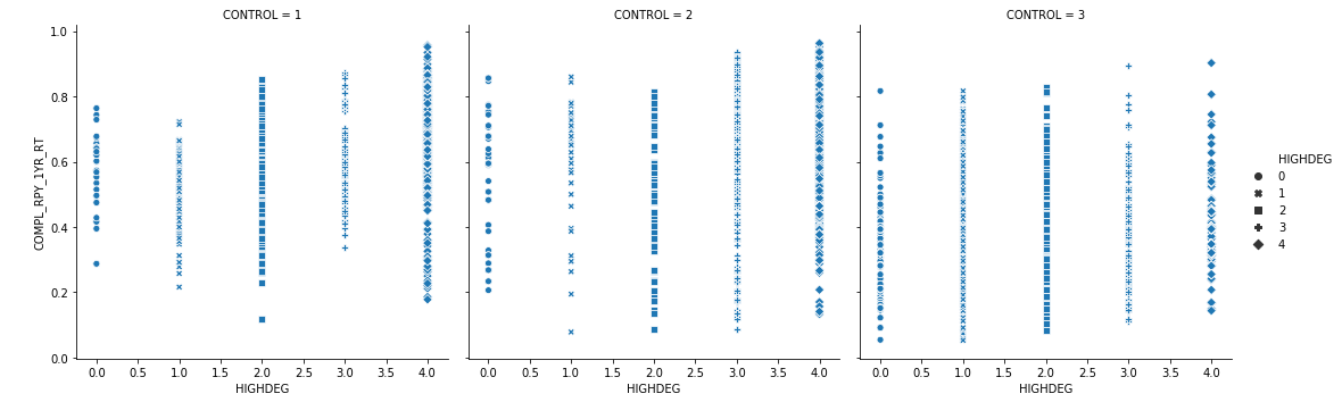
|  |
| --- |
| NONCOM\_RPY\_5YR\_RT 0.923287  COMPL\_RPY\_5YR\_RT 0.916724  COMPL\_RPY\_3YR\_RT 0.898931  COMPL\_RPY\_7YR\_RT 0.873590  NONCOM\_RPY\_3YR\_RT 0.869925  COMPL\_RPY\_1YR\_RT 0.840605  NONCOM\_RPY\_1YR\_RT 0.812678  LOAN\_DEATH\_YR3\_RT 0.733568  MD\_EARN\_WNE\_P10 0.691634  MN\_EARN\_WNE\_P10 0.690164  SAT\_AVG\_ALL 0.683118  ACTMTMID 0.670992  ACTCMMID 0.667276  SAT\_AVG 0.667223  AVGFACSAL 0.660287  PCTPELL 0.658074  SATMTMID 0.649815  ACTENMID 0.631715  COMP\_ORIG\_YR2\_RT 0.555561  HIGHDEG 0.553173  TUITIONFEE\_OUT 0.552523  CONTROL 0.525132  AGE\_ENTRY 0.505059 |

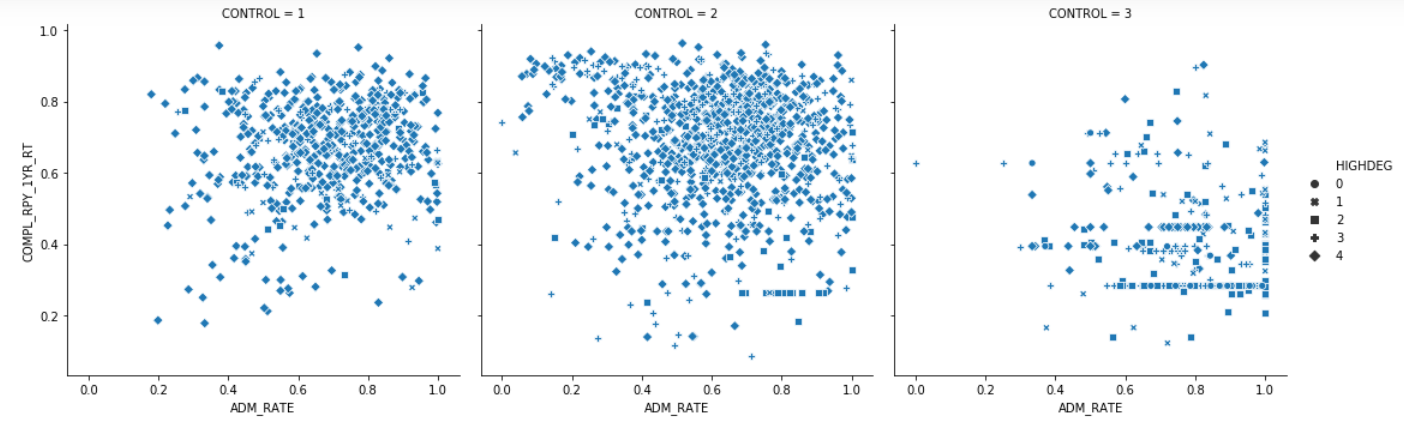
* Since the Correlation of each target with input features is very close to others, **COMPL\_RPY\_1YR\_RT** will be chosen as it is less risky, and drop all other candidates from the loaded dataset.

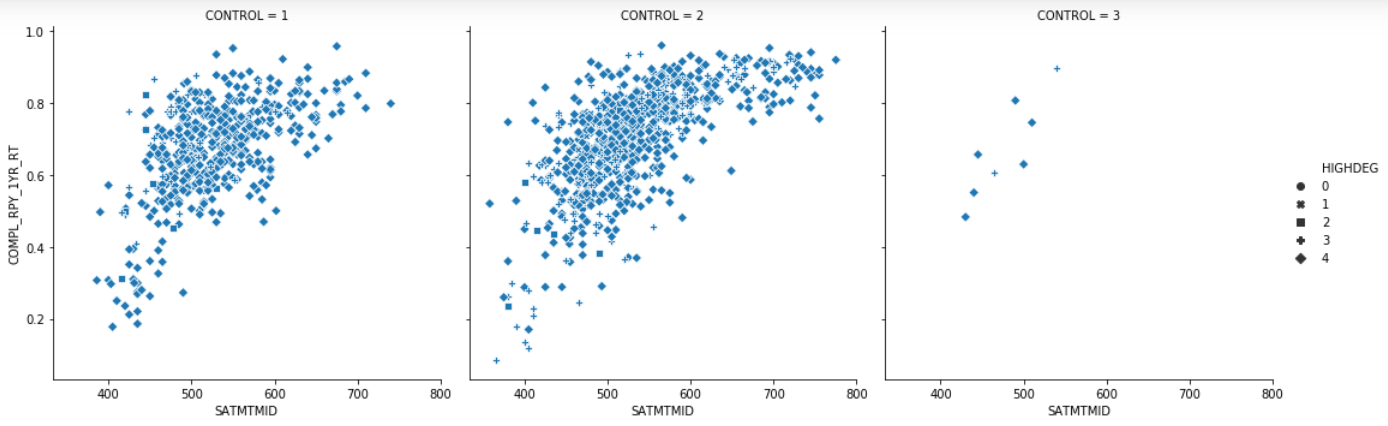


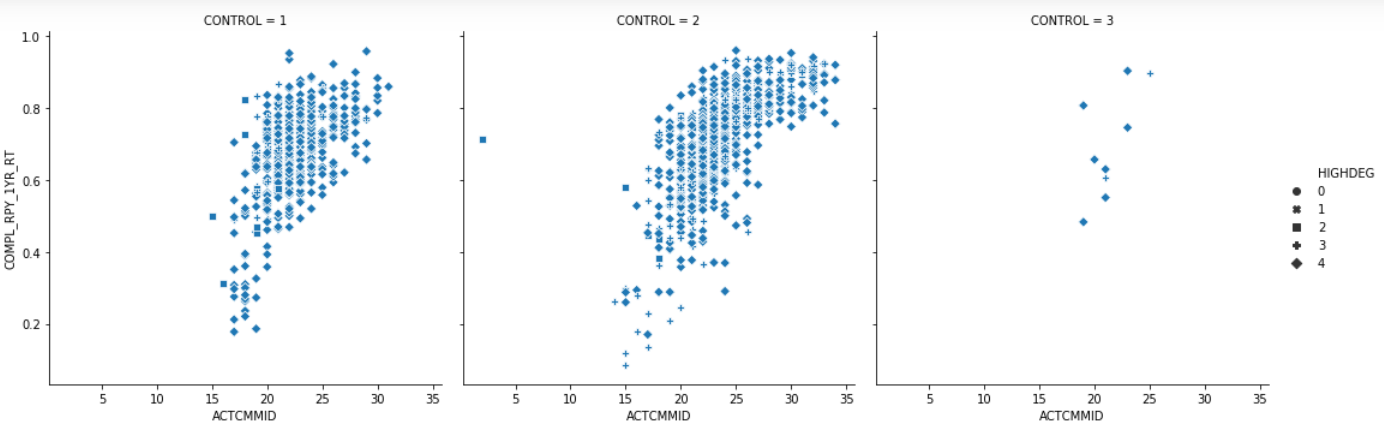


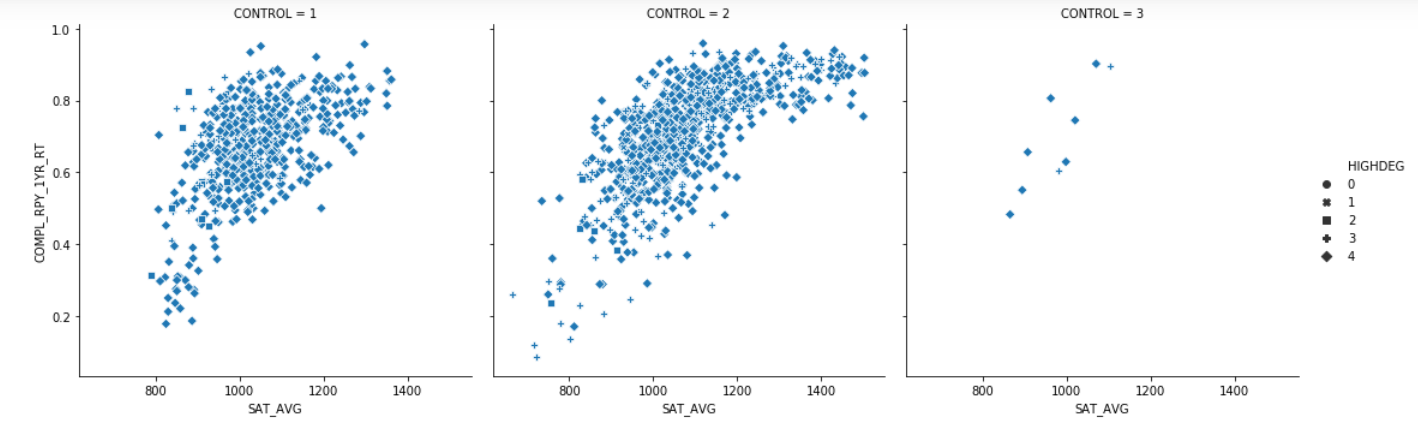
* Then visualize the relationship between the target variable and every input variable to check for any non-linear pattern

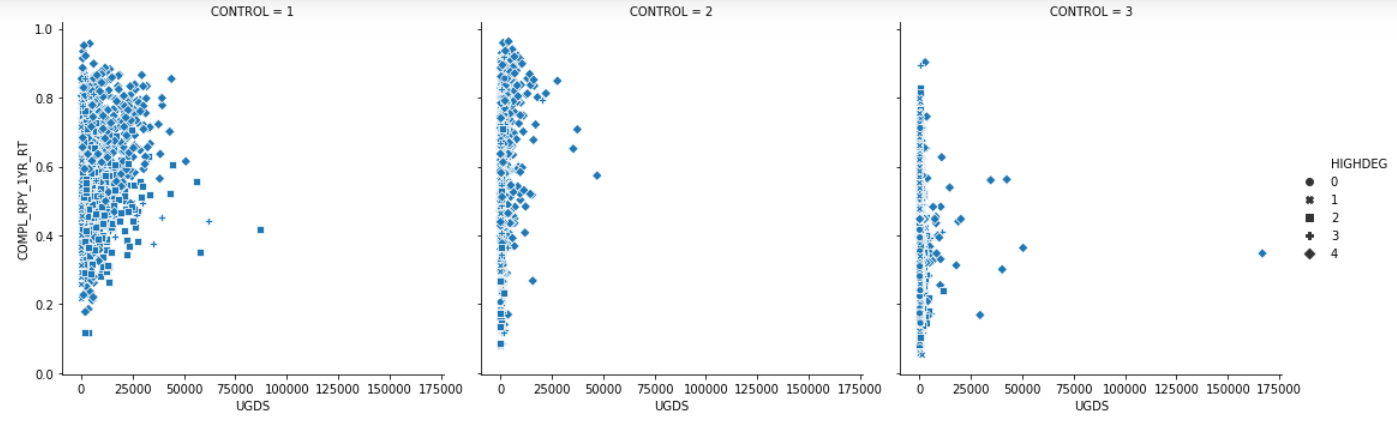


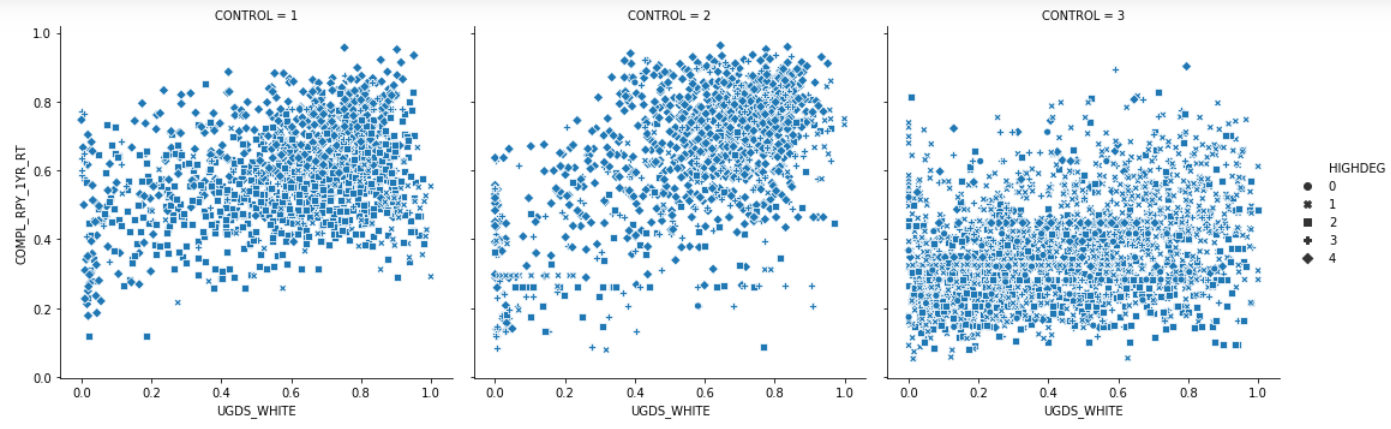


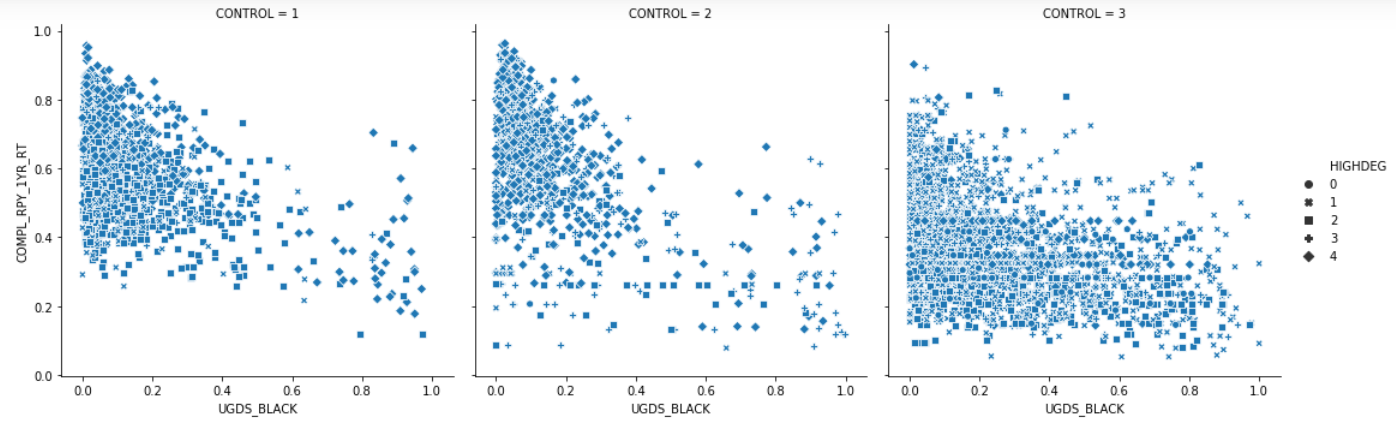


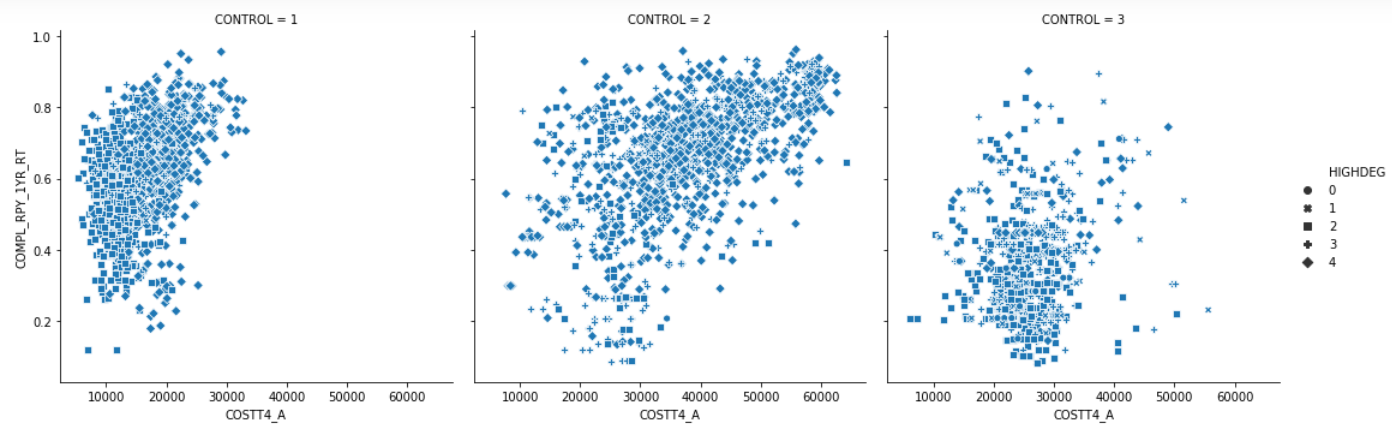


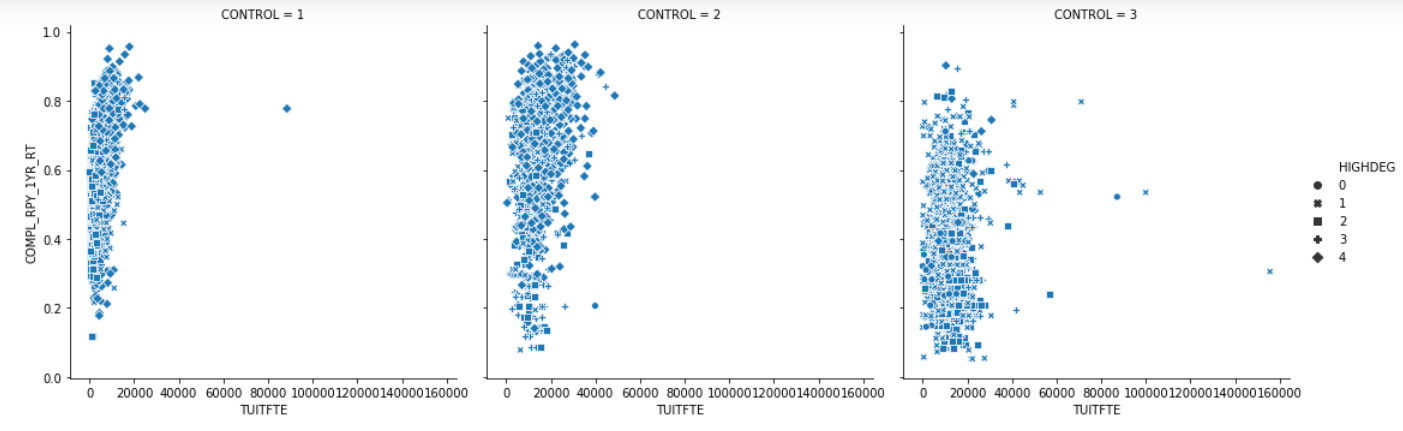


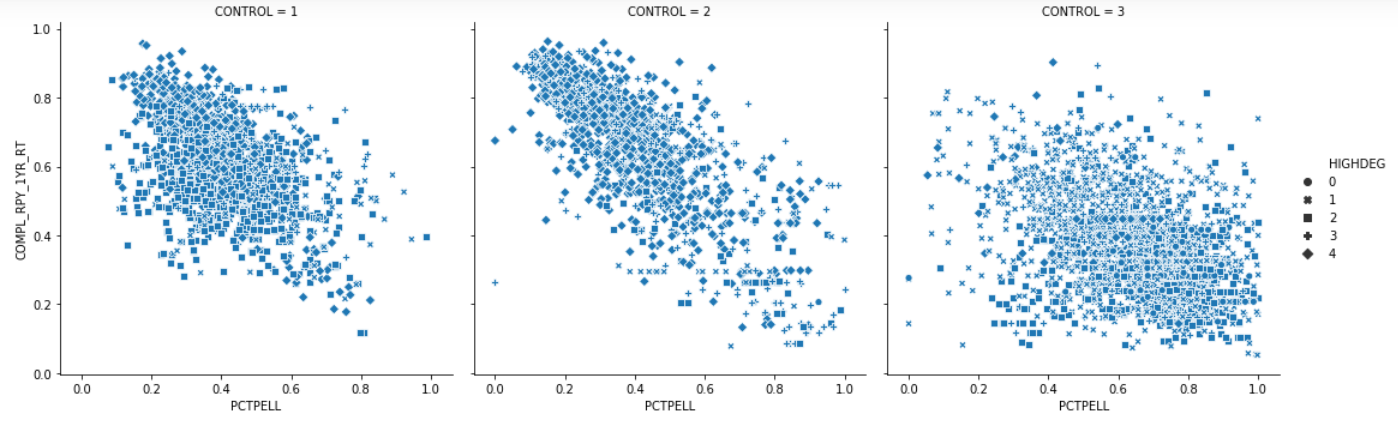


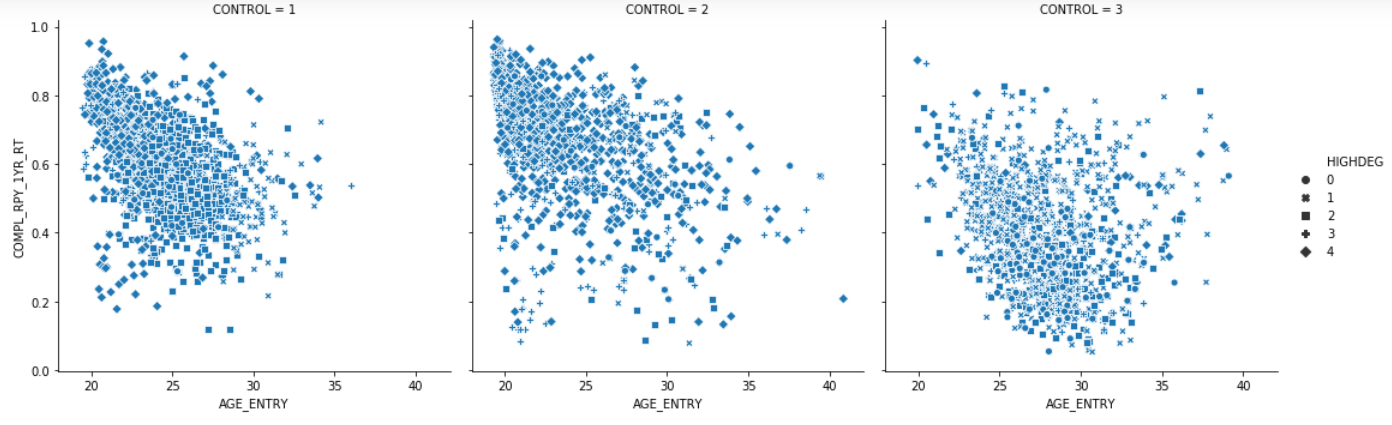


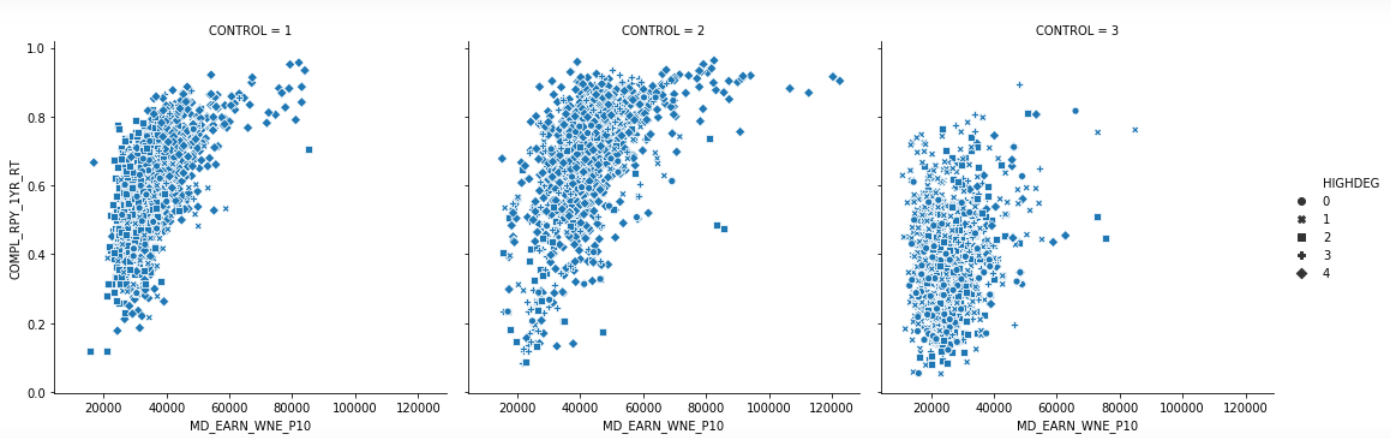








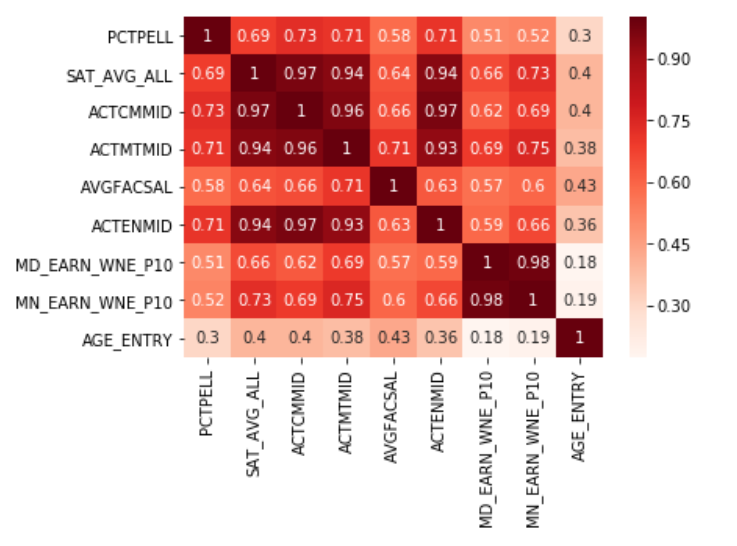




* Input features candidates:

|  |
| --- |
| PCTPELL 0.697547  SAT\_AVG\_ALL 0.668902  ACTCMMID 0.636761  ACTMTMID 0.628290  AVGFACSAL 0.607634  ACTENMID 0.605319  MD\_EARN\_WNE\_P10 0.591054  MN\_EARN\_WNE\_P10 0.590682  AGE\_ENTRY 0.581915  CONTROL 0.547110 (not a float column) |

* Then we checked the correlation between input features to see the relationship between input features.



## Cleaning Data

* Replace all "Privacy Suppressed" values with null value across all features
* Drop all rows that do not have a value for the target variable
* Preprocess the input features, By:
  + - Dropping the un-needed columns features (Index **INSTNM** and target output **COMPL\_RPY\_1YR\_RT**)
    - Splitting the input features into two types of float features and categorical features as these two types will be treated differently in preprocessing.
    - Encode categorical features as a one-hot numeric array, categorical features are (**CONTROL**, **HIGHDEG**)
* Splitting the data to the training set (80%) and testing set (20%)
* Using SciKit Simple Imputer to replace missing or null values with mean/median values
* Building a pipeline of preprocessing

# Modeling

## Selecting Modeling Techniques

* According to the nature of our data and our objective which is predicting the loan repayment probability, we select to start with the following five regression models:
  + XGBoost
  + Ridge
  + Lasso
  + Multiple Linear Regression
  + SVR ( with two kernels : Linear , and RBF )
* Since we have a lot of missing values and outliers, it is preferable to start with XGboost.

XGboost algorithm is invariant to outliers and can handle missing values by default, so first strategy is replacing any wrong/missing value by null, build the model and check the accuracy.

## Generating a Test Design

* The criteria by which the models are assessed is the following:
  + Calculating MSE, RMSE and r2 values and considering the model that has lower RMSE and higher r2.
  + Explore feature importance for each model and revising it against the initial findings for correlation and P-Value.

## Building the Models

### Parameter Settings

* models were built first using their default parameters, then after revising the error and based on the case ( overfitting/ underfitting) we used Grid Search technique to get the best parameters for each model.
* The chosen error function is squared\_error to be sensitive to higher errors.

## Assessing the Model

* Chosen evaluation criteria were used to compare different models to choose the best. The table below is the results we got for each model.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | MSE Values | | r2 score Values | |
| Training Set | Tet Set | Training Set | Test Set |
| XGBoost | 0.0053 | 0.0075 | 0.8629 | 0.8045 |
| Ridge | 0.0101 | 0.0098 | 0.7388 | 0.7498 |
| Lasso | 0.0210 | 0.02083 | 0.4586 | 0.4716 |
| Multiple Linear Regression | 0.0100 | 0.0097 | 0.7425 | 0.7525 |
| SVR Linear | 0.0354 | 0.0355 | 0.0883 | 0.0985 |
| SVR RBF | 0.0079 | 0.0394 | 0.7950 | -0.0004 |

* Models error matrix after applying Grid Search and using the best parameters values

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | MSE Values | | r2 Values | |
| Training Set | Tet Set | Training Set | Tet Set |
| XGBoost | 0.003 | 0.006 | 0.92 | 0.834 |
| Ridge | 0.0100 | 0.0098 | 0.7428 | 0.7513 |
| Lasso | 0.0101 | 0.0097 | 0.7401 | 0.7527 |
| Multiple Linear Regression | 0.0100 | 0.0097 | 0.7425 | 0.7525 |

* XGBoost has been chosen as the best model based on the analysis above. After that we analyzed important. feature importance sorted in descending order for this model is as below:

|  |
| --- |
| [('CONTROL\_Private for-profit', 0.49292025),  ('SAT\_AVG\_ALL', 0.2238546),  ('PCTPELL', 0.05530315),  ('AGE\_ENTRY', 0.033805005),  ('MN\_EARN\_WNE\_P10', 0.023122836),  ('MD\_EARN\_WNE\_P10', 0.023000425),  ('UGDS\_BLACK', 0.01871721),  ('COUNT\_NWNE\_P10', 0.015959097),  ('ADM\_RATE\_ALL', 0.014765438),  ('UGDS\_WHITE', 0.013748974),  ('COMP\_ORIG\_YR4\_RT', 0.00748397),  ('COMP\_ORIG\_YR2\_RT', 0.0066271634),  ('PPTUG\_EF', 0.0061399858),  ('INEXPFTE', 0.005556901),  ('SATMTMID', 0.005271441),  ('TUITIONFEE\_PROG', 0.005024421),  ('TUITIONFEE\_IN', 0.0049120053),  ('COMP\_ORIG\_YR3\_RT', 0.0046591046),  ('TUITFTE', 0.0040798658),  ('HIGHDEG\_Associate\_degree', 0.0029162674),  ('LOAN\_COMP\_ORIG\_YR3\_RT', 0.0025635613),  ('UGDS', 0.0025409288),  ('TUITIONFEE\_OUT', 0.0024787188),  ('UGDS\_ASIAN', 0.0023418262),  ('CONTROL\_Private\_nonprofit', 0.0021285913),  ('COSTT4\_P', 0.0021087895),  ('AVGFACSAL', 0.0020022623),  ('COUNT\_WNE\_P10', 0.0019879332),  ('PCTFLOAN', 0.0017065941),  ('UGDS\_HISP', 0.0016695685),  ('DEATH\_YR4\_RT', 0.0015539941),  ('CONTROL\_Public', 0.0015435874),  ('UGDS\_NHPI', 0.0015275691),  ('UGDS\_UNKN', 0.0014071261),  ('UGDS\_NRA', 0.001291639),  ('UGDS\_2MOR', 0.0011405724),  ('COSTT4\_A', 0.0009246818),  ('ADM\_RATE', 0.0008819474),  ('UGDS\_AIAN', 0.00033204918)] |

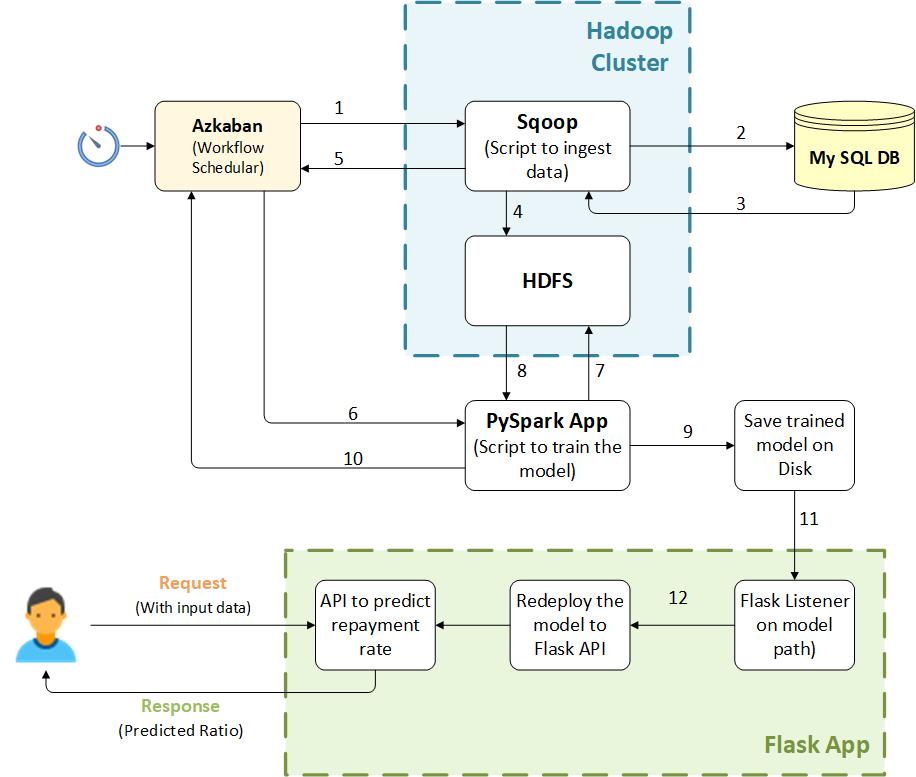
# Deployment

## System Main Components

* My SQL Database: Where raw Scorecard data for US institutions was stored.
* Azkaban: Is a batch workflow job scheduler created at LinkedIn to run Hadoop jobs. It resolves the ordering through job dependencies.
* Sqoop: Apache application used for transferring data between relational databases and Hadoop HDFS for MapReduce processing and so on.
* HDFS
* Spark App.
* Flask: A web framework provides some tools, libraries and technologies that allow to build a web application in python.

Flask application is a portal where end user can submit his request and show the response

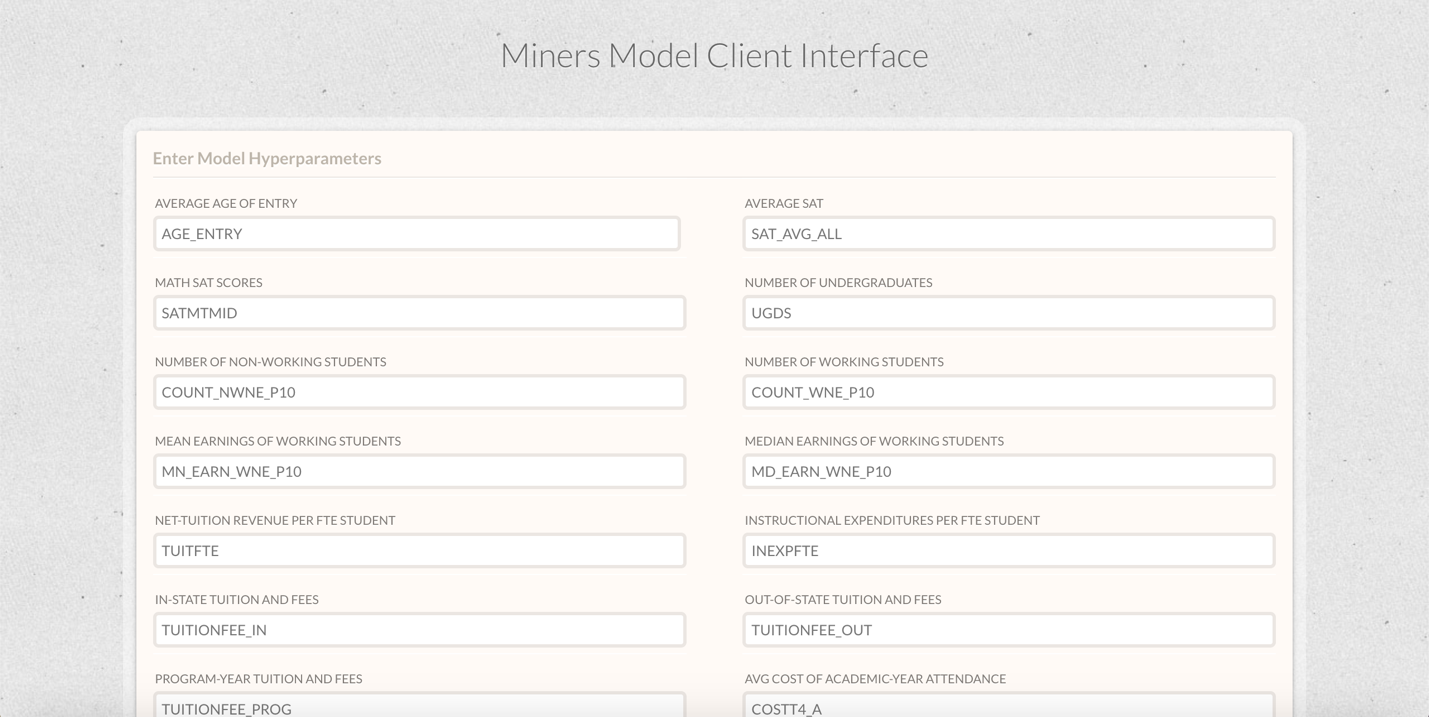
## System Workflow



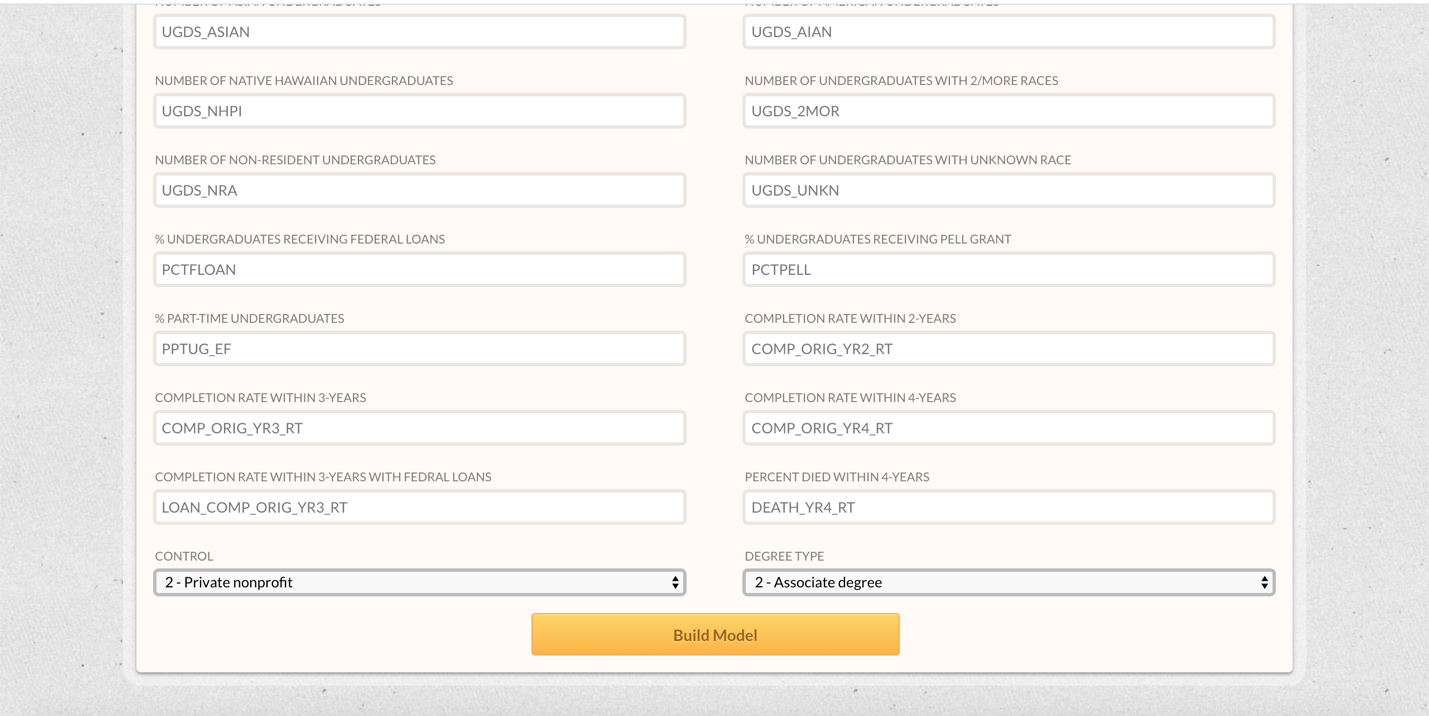
* 1: Azkaban scheduler triggers Sqoop script every 15 min
* 2: Sqoop script starts to ingest chunk of data from My SQL RDMS
* 3: Return data chunk to Sqoop
* 4: Sqoop save returned data to HDFS
* 5: Sqoop acknowledge Azkaban that its task is completed
* 6: Then Azkaban scheduler triggers PySpark App to start data processing task
* 7 , 8: PySpark App executers drill data from HDFS , and execute training jobs using the best parameters for XGBoost model
* 9: PySpark App executers save the trained model to the disk
* 10: PySpark App acknowledge Azkaban that its task is completed
* 11 , 12: Flask listener listen to the trained model updates saved on the disk , Then redeploy the updated model to Flask API

# End User App

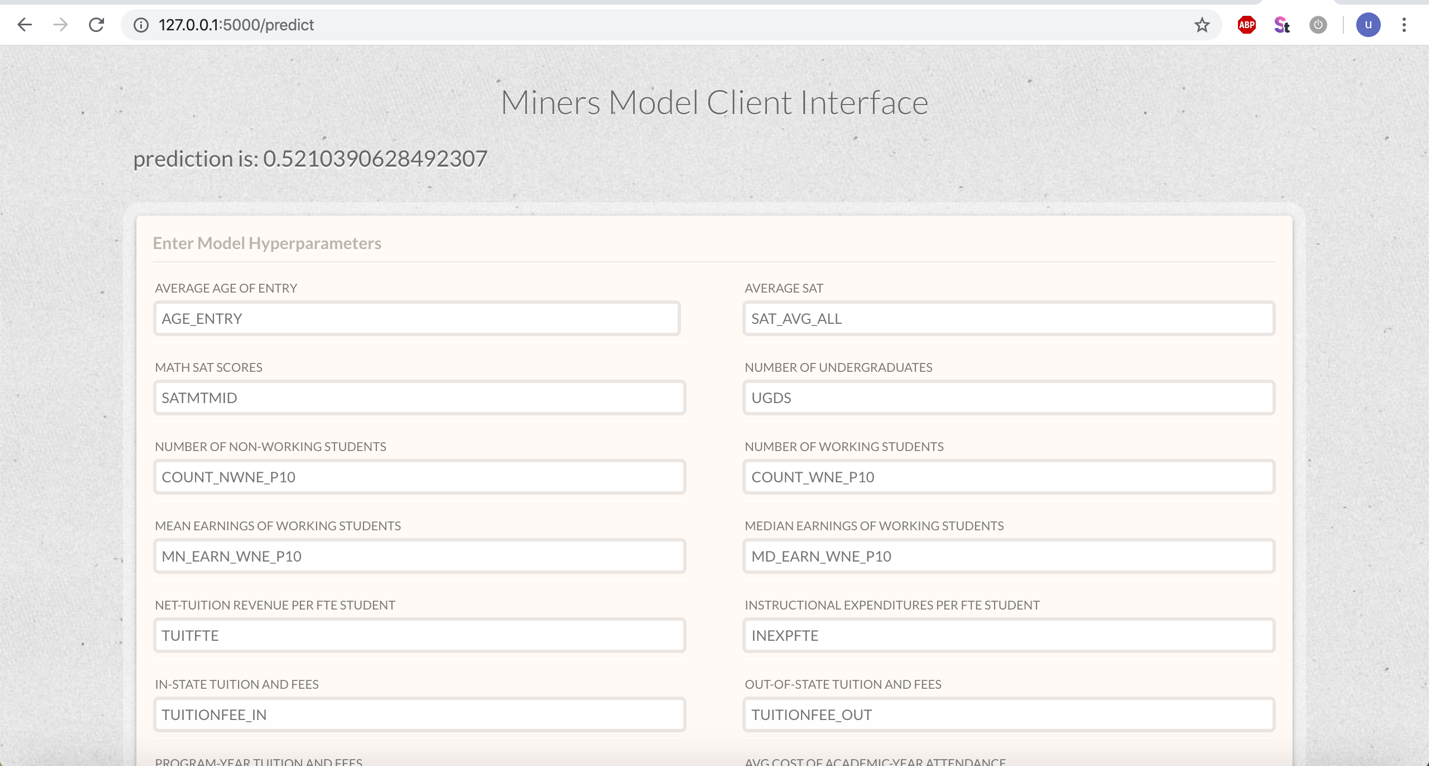
The interface of the end user app is as below.



The user fills the information he has about the institute in question and press on Build Models.



Then he gets the probability of repayment for this institute (if a student took a loan and got enrolled in this institute, the probability of repayment is the result)



# References

* <https://collegescorecard.ed.gov/data/>
* <https://www.kaggle.com/kaggle/college-scorecard>
* <https://github.com/benhamner/us-college-scorecard>
* <https://www.kaggle.com/apollostar/which-college-is-best-for-you/report>
* <https://www.kaggle.com/apollostar/which-college-is-best-for-you-part2/report>
* <https://rpubs.com/random_Island/education-loan-repayment>
* <https://www.americanprogress.org/issues/education-postsecondary/reports/2016/12/19/295187/sharing-the-risk/>