Website accessible at https://cs3244-project.herokuapp.com/

Prediction of Student Earnings and Loan Repayment Rates

Authors: Amrut Prabhu, Andres Rojas, Ang Jing Zhe, Eeshan Jaiswal, Sreyans Sipani
Emails: e0139127@nus.edu.sg, e0368325@u.nus.edu, e0030990@u.nus.edu, e0367841@u.nus.edu, e0238583@u.nus.edu

Abstract

This project aims to look at aggregate student characteristics in US colleges to build a machine learning model to estimate the predicted earnings and loan repayment rates of a given student. In order to get the most accurate results, we experimented with different models and analysed the factors leading to these results. This project is significant because nearly 40 percent of people who have taken student loans are expected to default on them by 2023, which will increase the annual number of defaulters from the current 1 million [7]. By allowing students to use our webpage to see their predicted earnings and loan repayments rates, we aim to help students with their financial planning so that they can make an informed decision about their choice of college. One of the highlights of the project is that our best model, Gradient Boosting with Regressor Chain, is able to get up to 90.76% accuracy on the entire dataset. From this model, we found out that the characteristics that have the most impact on the future earnings and repayment rates of students are their household income, loan amount and the tuition fees of the college.

Motivation and Importance of the project

The promise of a college education leading to a significant increase in earnings compared to a high school diploma [4] has resulted in more people wanting to pursue a college degree. Many of them end up taking student loans, but some are unable to repay them and end up defaulting the loan.

The number of students in default on student loans in the U.S reached 5 million in 2017 [5], with 40% of borrowers expected to default by 2023 [7]. With the amount of student debt at USD 1.4 trillion in the U.S. in 2018 [4], there is no question that this is a problem that needs to be addressed. In fact, it makes up the largest portion of U.S. non-housing debt. This is the motivation for our project.

Dynarski [3], an economist, had published a paper in 2014 on her perspective on student loans in the United States. In her paper, one of the existing solutions that aim to reduce defaults is an income-based repayment system. However, she also state that a good repayment system requires data on individual earnings and borrowing.

A good first step towards reducing the number of defaults is to empower people who take student loans to have good financial planning. In order to do this, one of the key factors is to know the future earnings and loan repayment rates of the person. This motivated us to create machine learning models to predict these key factors since we had a lot of data to gain insights from. After training these models on the College Scorecard dataset [10], we built a novel online tool where students can input the required characteristics and get predictions of the aforementioned factors. With it, we hope that future financial aid takers will be capable of making informed decisions and lowering their chances of defaulting on their education loan.

As for our results, we concluded that a Gradient Boosting Regressor with a Regressor Chain is a good choice of model for this dataset. In addition, we found out the relative importance of different characteristics in determining the future financial state of a person. To come up with these results, we set out with the following guiding questions:

Goals:

1. What are the future earnings of a student based on tuition fees, location, family income, degree type, and so on?
2. What are the future repayment rates of a given student based on characteristics like family income, loan amount, university type and so on?

Stretch Goals:

1. Which type of model is most suitable for our dataset and why is it so?
2. What are the insights gained from analysing the results and the features involved?

In this report, related work in this topic and on the dataset is discussed to give some background information. Next, the dataset is described along with the methods that were employed to process the large amount of data into a format that could be understood by us and our models. This is followed by a description of the models and methods that were used to learn from the dataset. Next, the macroscopic results of the various models are shown and compared. Finally, there is a discussion about our results and insights gained about the entire process, before proceeding to conclude the report.

Related Work

Similar research on college education defaults have been done with a different focus and approach. A lot of work done on the College Scorecard dataset is statistical, but there has been some machine learning work done on it as well. We mainly looked at three reports and decided to learn from their strengths and mistakes.

Dean [2] studied the same dataset as the one we have chosen. He started off with a hypothesis about the characteristics of a better school and performed a lot of data analytical to verify his hypothesis. He narrowed down the features to those that influenced future earnings and with the reduced set of features, he built a supervised regression model to predict future earnings. We gathered from his analytical results that there are a multitude of insignificant features in the dataset and hence, we should do feature selection carefully for the models in our project.

Govindarajan [6] also worked on the College Scorecard dataset. However, the focus was different- it was to explore methods of imputations for missing data since the dataset has a lot of invalid values. Following this, they built their model with ridge regression and random forest. We realized that a gradient boosting algorithm would perform better and were able to achieve an RMSE score of ~6000 for our earnings target values. With their work in showing how imputations can still maintain good accuracy, we are confident in our approach for handling missing or invalid data in our dataset.

A group of students from Stanford university also worked on this dataset and had the same focus as our project [1]. However, their data pre-processing technique, which is to select only 1 year as the data, is different from what we did. In addition, they built various models such as linear regression, kNN regression, SVM and neural network. While they seem to hit good accuracies, they removed a lot of values from their dataset.

What we also realized is that most of the current work focuses on earnings only. But, we believe that repayment rates are equally or even more important when it comes to understanding if one is able to pay back their loan or not. Therefore, we spent some time trying to accurately predict these values as well.

The dataset we chose is very vast and a good understanding is required to actually analyse the results of the models. Some of the reports do not dig deeper into why certain results are obtained due to a lack of understanding of the dataset. We spent quality time preparing it and pruning it to obtain meaningful results.

Dataset

We worked on the College Scorecard dataset [10], which provides insights into the performance of US institutions that receive federal financial aid dollars, and the outcomes of the students of those institutions. The data is split into various CSV files that contain the collected results across the years for all the specified institutions. The files are- Most recent cohort data, National Student Loan Data System (NSLDS), and Post school earnings. One of the main downsides to this dataset is there is a considerable amount of missing values.

Data Merging and feature selection

Though the dataset contains multiple files, ideally, we would want to work with a single CSV file that contains all the required features. In order to do this, we merged all the data into a single CSV on an institution by institution basis. This will also lead to better results for filling in the missing values. This is because computing them using the existing data for a specific college across the years will provide better results than computing values for the data across all colleges for a particular time period (mean of a column across all colleges), as the missing values are more closely correlated to a specific college's environment, rather than the country as a whole.

After this, we worked on reducing the number of features (originally over 1500 columns). Upon further inspection, we realised that many of the columns are insignificant for the purposes of our project. For example, the breakdown of students per degree for each university, or the disaggregated data for features like median earnings. We picked **10 input features** to predict earnings and repayment rates: median debt, unemployment rate, state, household income, predominant degree, college type, tuition fees, poverty rate, currently operating and locale.

Some other features such as **stream** and **race** proportions at the college were also kept. This data was used to compute the dominant stream and race in the college. These columns could be incorporated if we considered each university instance as a student, which is acceptable looking at the size of the dataset, i.e., **9725 rows**. Therefore, we trained our model with and without these columns to see their effect.

Categorical and missing values

In order to not compromise on the accuracy of our regression models, we had to ensure that they appropriately dealt with the categorical values in our dataset. For this reason, we utilised a one hot encoder which makes it seem as though there are no categorical features in our data. Keeping in mind that increased dimensionality has its downsides, we made sure that our feature selection accounted for this. For example, we chose to keep the state variable instead of the city variables, as the latter one will greatly increase the total number of features, while not providing that much more.

To illustrate one hot encoding, consider the example of the *state* variable which previously made use of one column to indicate the state of a college, for example, CA for college in California. After one hot encoding, the state column is split into as many columns as there are categories, and now for the previous case, the value in column stateCA will be 1.

While this will increase the number of features, we believe it is important to allow our model to learn from our chosen categorical values as it is clear that where you go to college or where you end up working is a deciding factor on how much money you earn and spend.

Finally, we performed multiple imputation by chained equations (MICE) [12] to solve the aforementioned issue of missing values. Unlike other imputation methods which do not account for uncertainty in imputations, MICE does and produces results which are not overly precise. In addition, MICE allows for flexibility when performing the computations; it does not need to assume all the features follow a normal distribution. All these advantages make it a good fit for our problem at hand.

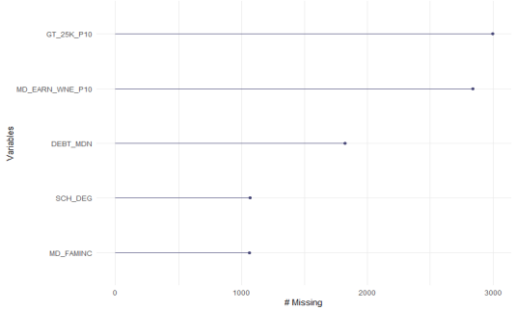


Figure 1. Number of missing values for some of our selected features in the College Scorecard dataset.

Figure 1 shows the number of missing values in the dataset before imputed values. Many features such as median earnings and family income are significant enough to not drop even if they are missing in ~20% of the cases. A more comprehensive graph is shown in Figure 7 in the Appendix. To understand the graph better, it might be helpful to have a look at the data dictionary to understand the features [11].

Target values

We have 7 target values from the dataset that are to be predicted. They are:

1. Median Earnings after 6 years: MD_EARN_WNE_P6
2. Median Earnings after 8 years: MD_EARN_WNE_P8
3. Median Earnings after 10 years: MD_EARN_WNE_P10
4. Repayment rate after 1 year: RPY_1YR_RT
5. Repayment rate after 3 years: RPY_3YR_RT
6. Repayment rate after 5 years: RPY_5YR_RT
7. Repayment rate after 7 years: RPY_7YR_RT

We believe that these 7 values will be enough for a student to understand his financial status in the future after university. It is also possible to derive a mathematical formula involving these values to determine whether you are predicted to default on your loan or not.

Methodology

Baseline model

We decided to test how a standard linear regressor would perform on the given dataset as there might be linear correlations between features like net price of the institution, household income and debt. We managed to get an accuracy of ~80% on our earnings target values and ~58.0% on our repayment rates. We decided to try some more complex models in order to obtain better results.

Random Forests and Gradient Boosting

We decided to use tree-based algorithms as we had a mixture of categorical and numerical features. We tested our dataset on 4 models, each used to predict the earnings and the repayment rates separately:

1. Multiple Output Regressor using Random Forests
2. Multiple Output Regressor using Gradient Boosting
3. Regressor Chain using Random Forests
4. Regressor Chain using Gradient Boosting

Here is a description of the techniques involved:

Random Forest Regressor:

1. We decided to make a forest of 1650 Decision Tree Regressors and then aggregate their predictions. We saw that as the number of estimators increased, the predictions become better up to a certain saturation point after which the number of trees does not make a difference. Increasing the number of trees also comes with a performance cost and therefore 1650 seemed to be the ideal number for the trade-off.

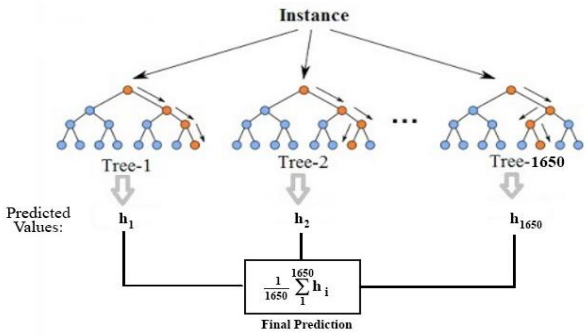


Figure 2. Visualisation of one of the Random Forests for Earnings prediction. Diagram credit: Wikipedia (modified)

2. The Decision Tree Regressors are created with the following parameters:
   a. A max depth of 80 was chosen as it was observed that increasing it further did not improve the accuracy and putting high values would have led to overfitting. This number was a maximum and we suspect that the trees rarely reached such a high value.
   b. We decided that at least one-tenth of the number of samples were required to make a split. We know that as we decrease this fraction, the number required to split is lesser, leading to more splits and terminal nodes with lesser sample data. This leads to a performance issue and also overfitting. A value of 0.1 is optimal.
   c. The default configuration of taking square root of number of features as maximum features for a tree works fine and trying other configurations such as logarithm of number of features or number of features did not provide any significant improvement. Therefore, we decided not to change it.
   d. We decided that when subsamples are being made for the different trees, it will be done so without replacement due to a high number of estimators.
3. All the trees are then built in a parallel fashion as they are independent of each other.
4. When the random forest has to predict, it takes the average of the predictions of all the trees in the forest.

Gradient Boosting Regressor:

1. The Gradient Boosting algorithm starts by making an estimator that predicts the median value for the target.
2. We decided to use Least Absolute Deviation for our loss function as it prevents overfitting and reduces susceptibility to outliers, unlike the Least Squares Error loss function.
3. The residuals are calculated based on this loss function.
4. A tree is built to fit the residuals obtained. Step 3 and Step 4 are repeated to build 1900 trees sequentially fitting the residual for the previous tree. Similar to the random forest, we noticed that the accuracy wasn't affected after a certain saturation point. Being a sequential algorithm, we did not choose a tremendously a high value as that would affect the performance drastically.
5. The Decision Tree Regressor are created with the following parameters:
   a. Similar to the decision tree in the random forest

b. regressor, a max depth of 70 was chosen as it was observed that increasing it further did not improve the accuracy much.
c. A learning rate of 0.1 is chosen which means the weightage of the subsequent trees are shrunk by 0.1 each. This can be represented by the formula below - $F_m(x) = F_{m-1}(x) + \alpha h_m(x)$ where h_m is the model represented by the m^{th} tree.
d. We decided to set a minimum value for samples that the leaf requires also. This overrides the minimum samples required to split criteria if it cannot find enough sampled for the left and right leaf. This makes sure no leaf has too less values which can lead to overfitting. We believe that this value should be equal to 1/2000 times the number of samples ~ 5 samples. The default value of 1 sample seems to be too less for an optimum model.
e. We decide to use the previous model or solutions to speed up the learning process.

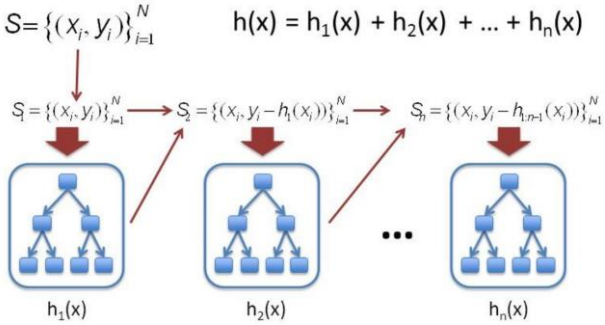


Figure 3. Visualisation of the Gradient Boosting Algorithm - We can see how each tree fits the residual value and the predicted value is a summation of the prediction of all the trees. Diagram credit: Dimensionless

6. All the trees are built sequentially.
7. When the model has to predict, it takes summation of all the values predicted by each tree and returns it as output.

As we decided to predict multiple target values, we decided to explore two simple algorithms for this - the Multiple Output Regressor and the Regressor Chain.

Multiple Output Regressor:

The Multiple Output Regressor is equivalent to building individual regressor models for each target value. Its advantage is a significant gain in performance as all the models can be built in parallel, independent of each other.

Regressor Chain:

The Regressor Chain decides to use the target values that have already appeared as a feature for the current target value as well. For example, if we have n features and m_1, m_2 … upto m_i as our target values. Then m_1 is trained with the n features, m_2 is trained with the n features + m_1 … upto m_i which is trained with n features + m_1, m_2 … m_{i-1}. The algorithm uses a stratified 3-fold cross-validation for the results of previous estimators in the chain. This elaborate setup impacts the performance.

Two main techniques were used sequentially for tuning the hyper parameters - Randomized Search and Grid Search.

Randomized Search

This is the first step used to estimate the correct values for the hyper parameters [8]. A list of values that must be tried for the hyper parameter is curated. A set of all the combinations is created such that the individual impact of each hyperparameter can be observed. For example - different combinations of the number of trees and learning rate are tried out (as there exists a trade-off between them) while keeping the other hyper parameters constant. We tested the model for about 300 - 600 combinations out of all possible combinations to narrow down a rough range for the hyper parameters. The scoring formula used to evaluate the models which perform better is the MAPE or Mean Absolute Percentage Error. Therefore, the error is defined as -

$$\mathrm{M} = \frac{100\%}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|$$

and the accuracy is defined as (100 - M)%. A similar stratified 3-fold cross-validation is applied to obtain a more accurate score.

Grid Search

This is the second step involved in determining the values of all the hyper parameters [8]. Here we exhaust all possible values which has been narrowed down by the randomized search and choose the model with the highest score as the best model. The hyper parameters for this model are analysed, reasoned and chosen. The scoring formula and cross validation step remains the same as above.

Results

Comparison of Model Predictions:

In order to gauge the performance gap between our 4 models, we used each of them to predict the median earnings and repayment rates. We then proceeded to use linear regression to see the trendline between the predicted and actual values of the output. The results are shown in Figure 4 below.

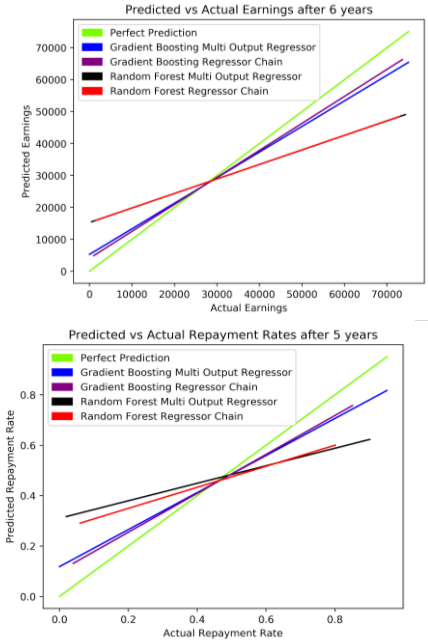


Figure 4. Comparison of the accuracies of the models on the test dataset.

It is evident that the Gradient Boosting Regressor Chain most closely follows the green line and hence, is the most accurate. Its accuracy is against the test set is documented below:

Target Value	Accuracy
Median Earnings after 6 years	85.73%
Median Earnings after 8 years	85.44%
Median Earnings after 10 years	85.29%
Repayment rate after 1 year	72.08%
Repayment rate after 3 years	76.27%
Repayment rate after 5 years	79.6%
Repayment rate after 7 years	82.35%

The above model also hit high accuracies (approx. 91%) on the whole dataset itself. This is partly expected as the Gradient Boosting Regressor is a boosting algorithm while Random Forest Regressor is a bagging algorithm. This is because the boosting algorithm sequentially tries to fit the errors made by the previous tree and improves after every iteration. The Random Forest just takes the average of the predictions given by its trees and therefore has a higher chance of error.

The Regressor Chain was chosen over the Multiple Output Regressor as it was seen that it performed better with the Random Forest Regressor. This gave us the intuition that the Regressor Chain is able to learn some correlation between the target values itself and therefore we decided to use it for more accurate results. This can be shown by looking at the accuracies for both when used with the Random Forest Regressor.

Target Value	Multiple Output Regressor	Regressor Chain
Median Earnings after 6 years	80.02%	**80.04%**
Median Earnings after 8 years	79.35%	**79.95%**
Median Earnings after 10 years	78.95%	**79.73%**
Repayment rate after 1 year	58.96%	**58.97%**
Repayment rate after 3 years	66.21%	**68.36%**
Repayment rate after 5 years	70.98%	**73.35%**
Repayment rate after 7 years	75.93%	**77.26%**

Discussion

Feature Importance

To calculate the importance of every feature, we decided to use the 'gini importance' or the mean decrease impurity of the features. It is defined as the ratio between the number of splits that include the feature to the number of samples split by the feature averaged over all the trees. We can clearly see that important financial features such as household income (MEDIAN_HH_INC), average net price of the institution (NPT4_PUB) affect the model's prediction the most. Due to one-hot encoding, it is hard to measure how much the categorical variables affect but we can make inferences such as attending university in Idaho (STABBR_ID) or choosing to study business marketing (STREAM_PCIP52) affects the model's predictions more than the other categories. Access the data dictionary [11] for a full description of each feature.

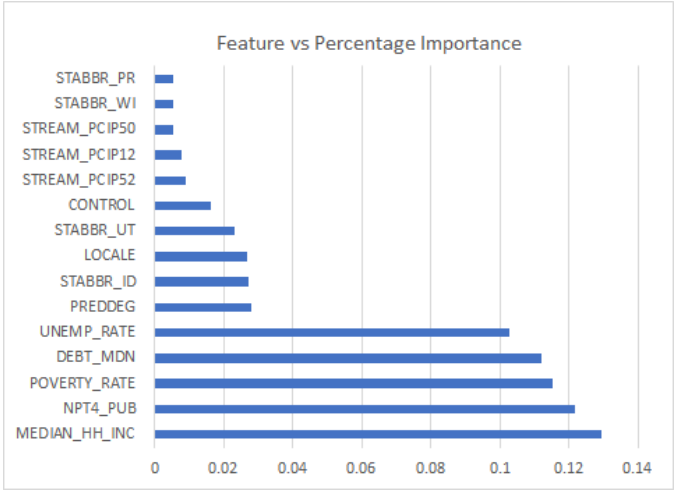


Figure 5. Graph showing the importance of features (after one-hot encoding) in our model (top 15 shown).

Predictions on the whole dataset

The predictions on the whole dataset deviate much more from the ideal green curve. This may be due to:

- In the real world, it may actually be accurate. Most people don't earn exorbitant sums of money. The College Scorecard dataset contains aggregate values.
- It may be due to the fact that we used Least Absolute Deviation as a cost function instead of Least squares. Hence outliers are punished as much, and our model underfits to them.

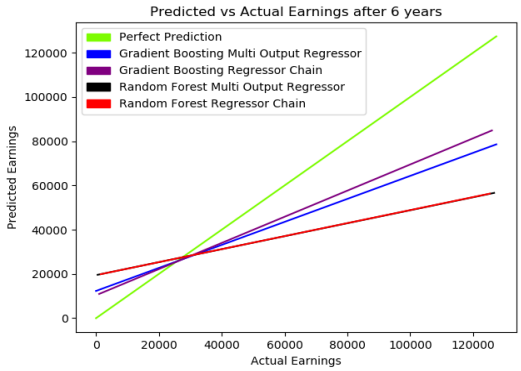


Figure 6. Comparison of the accuracies of the models on the whole dataset

To understand this further, we decided to analyse it better by looking at the top 1004 rows with the highest error. On

analysing all the columns we see that the values of debt are 39.13% (on an average) off the mean and the net price of the institution was 36.89% (on an average) off its mean. We understand that the model fails to predict when these values start becoming extreme. For example, certain rows of the dataset portray universities with low net price but high future earnings. The model is not able to fit these universities. While intuitively it seems difficult to fit such institutions which are able to give good returns with low fees and price, we must try to understand these universities do so and maybe find features that need to be added so that the model can fit such special cases also.

Intersection points of models

Looking at the graphs for earnings after 6, 8 and 10 years, the intersecting points of the models may be indicative of the fact that our model is extremely accurate in cases involving the average person. This may be a consequence of the fact that the data used for training the model represented aggregate data. However, the points of intersection are increasing over time, as is to be expected.

Correlation between repayment rates

As mentioned before, using a regressor chain shows us that the repayment rates in future years are dependent on the repayment rates for previous years. But the same is not true for the earnings over the years. This is interesting as we would expect these to be closely related but both do not follow the same trend in this case.

Effect of adding other features

One of the ideas that set us apart from previous work and other projects was our motivation to build a tool people could use rather than just analysing the factors affecting future earnings. While this was difficult to achieve with the dataset used, we first decided to build a model based on features a user can input. Keeping the above in mind, we finally also added the dominant stream as a feature to understand its effect on the performance of the model. The following statistics show that it is able to improve the model accuracy by about a percent for the earnings while interestingly the accuracy for repayment seems to decrease slightly. This is another sign about the difference in trends between the repayment rates and earnings against out models. The following table shows the accuracy (based on the MAPE formula) on the whole dataset:

Target Value	Without Stream	With Stream
Median Earnings after 6 years	90.76%	**91.39%**
Median Earnings after 8 years	89.93%	**90.52%**
Median Earnings after 10 years	89.08%	**89.64%**
Repayment rate after 1 year	83.15%	**83.3%**
Repayment rate after 3 years	85.25%	**84.95%**
Repayment rate after 5 years	86.33%	**85.96%**
Repayment rate after 7 years	87.56%	**87.32%**

Conclusion

Our project is able to help students learn more about how they are shaping their future. We decided to go one step further than other projects by projecting how they would be able to repay back their loans. For the sake of accurate results and a usable tool, we decided to select features more carefully and tune our models well rather than naively build models on the original dataset. However, we still believe that, in the future, we can add more attributes in a meaningful way such as student aptitude, which we were unable to do due to too many missing values regarding SAT scores and ACT scores (see Figure 7 in the Appendix).

Similarly, the university the student is going to is also not taken into account, so we are not able to make the model learn that better universities might lead to better earnings and repayment rates. As mentioned before, our model also did fail to cover some extreme cases and therefore we need to analyse and improve our model to fit such extreme cases using additional data.

During data preprocessing, we had also split our dataset into different sets that relate to a specific type of student, for example: female, male, high family income or low family income. We did not end up using these for this project. However, we believe these could be useful to build more tailored models and is a potential future improvement on our work. As we have data regarding earnings and repayment over several years, we plan to build time-series machine learning models to predict earnings on a long-term basis as well.

In conclusion, even though we are able to produce accurate results now, we plan to do much more with the dataset at hand in the future. We will also look into additional data to improve the interpretability and usefulness of our models.

References

[1] Agrawal, M. et al (2015). *Prediction of Post-Collegiate Earnings and Debt.* Available at: http://cs229.stanford.edu/proj2015/212_report.pdf

[2] Dean, Jason (2017). *College Scorecard Data Analysis and GBM Model of Predictive Earnings.* Available at: http://jasontdean.com/R/collegeScoreCard.html

[3] Dynarski, S. (2014). *An Economist's Perspective on Student Loans in the United States*. [online] Brookings.edu. Available at: https://www.brookings.edu/wp-content/uploads/2016/06/economist_perspective_student_loans_dynarski.pdf

[4] Maldonado, C. (2018). *Price Of College Increasing Almost 8 Times Faster Than Wages.* Forbes. Available at: https://www.forbes.com/sites/camilomaldonado/2018/07/24/price-of-college-increasing-almost-8-times-faster-than-wages/#1854bcad66c1

[5] Mitchell, J. (2017). *Nearly 5 Million Americans in Default on Student Loans*. The Wall Street Journal. Available at: https://www.wsj.com/articles/nearly-5-million-americans-in-default-on-student-loans-1513192375

[6] Govindarajan, N. (2017). *To College or Not to College: Prediction of Post Collegiate Earnings and Debts*. Available at: https://github.com/nithya4/CS-589-College-Scorecard

[7] Nova, A. (2018). *More than 1 million people default on their student loans each year.* CNBC—[Online]. Available at: https://www.cnbc.com/2018/08/13/twenty-two-percent-of-student-loan-borrowers-fall-into-default.html

[8] Scikit Learn. *Tuning the hyper-parameters of an estimator.* Available at: https://scikit-learn.org/stable/modules/grid_search.html

[9] Study (n.d.): *How Much More Do College Graduates Earn Than Non-College Graduates?* Study. Available at: https://study.com/articles/How_Much_More_Do_College_Graduates_Earn_Than_Non-College_Graduates.html

[10] US Department of Education. (n.d.). *College Scorecard Data*. College Scorecard —[Online]. Available at: https://collegescorecard.ed.gov/data/

[11] U.S Department of Education. *College Scorecard Data Dictionary.* College Scorecard. Available at: https://collegescorecard.ed.gov/assets/CollegeScorecardDataDictionary.xlsx

[12] van Buuren, S. and Groothuis-Oudshoorn, K. (2011). *mice: Multivariate Imputation by Chained Equations in R*. Journal of Statistical Software, 45(3) [online]. Available at: https://www.jstatsoft.org/article/view/v045i03/v45i03.pdf

Appendix

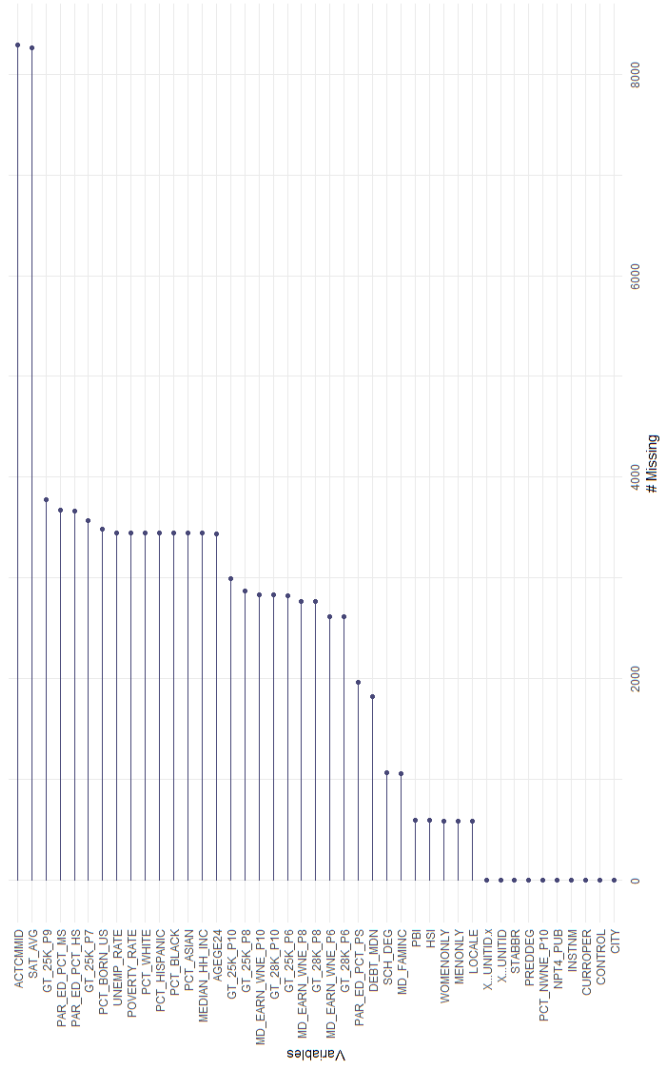


Figure 7. Number of missing values for features in the College Scorecard dataset (Highest numbers shown).

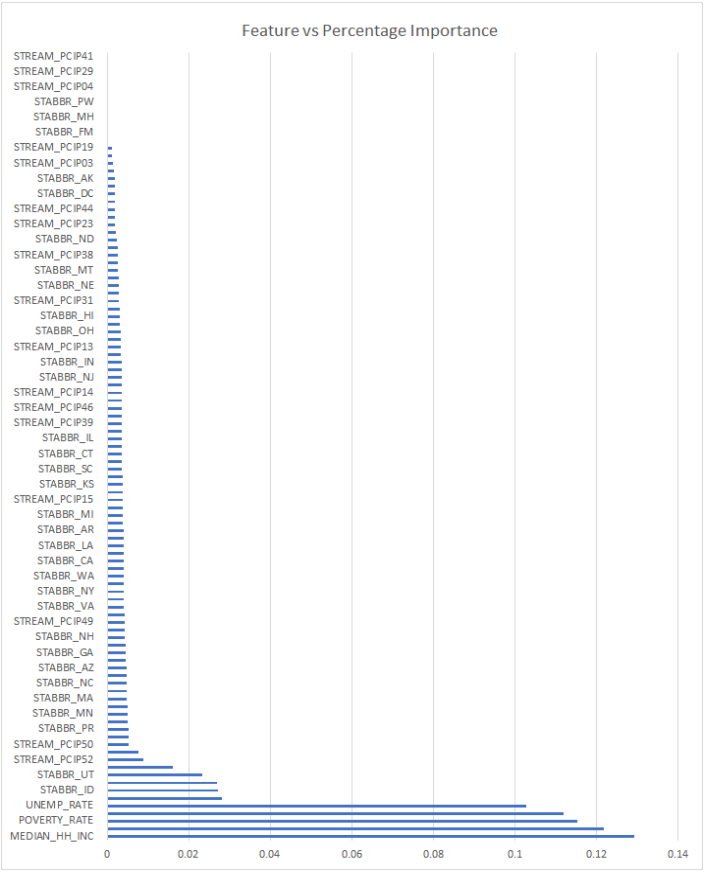
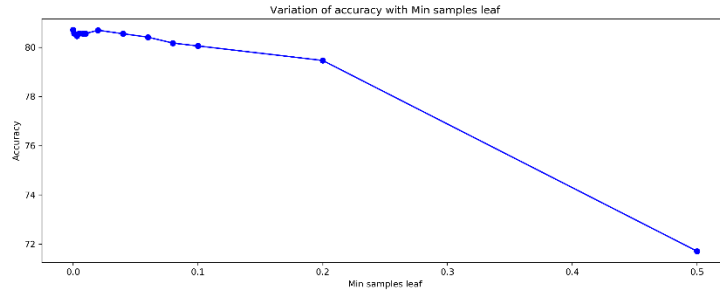
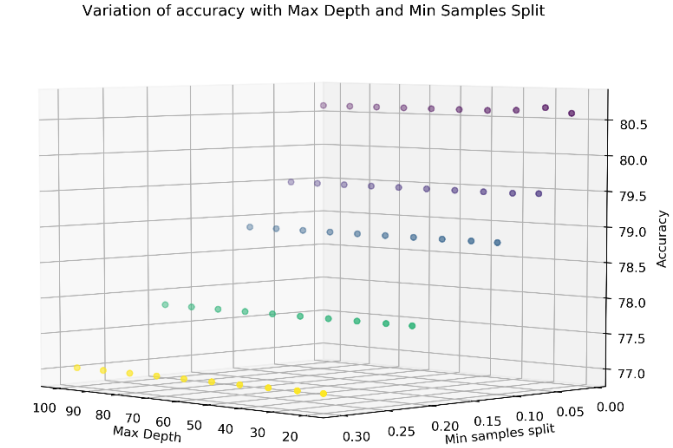
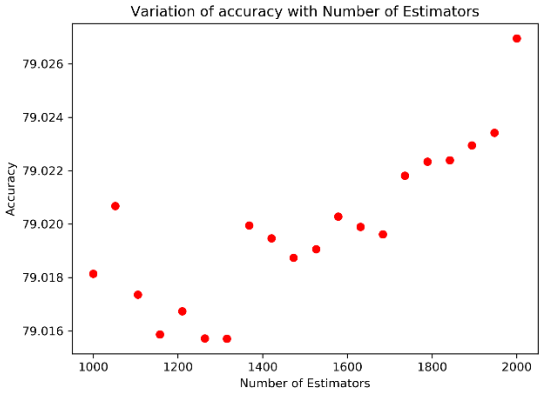


Figure 8. Graph showing the importance of features (after one-hot encoding) in our model.

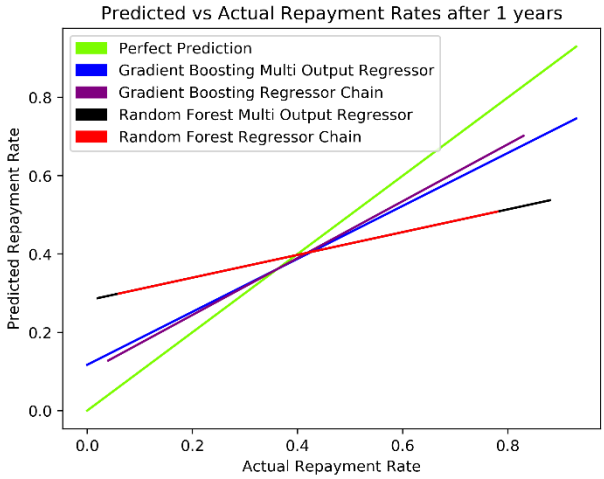
The following plots show the results of the parameter tuning of our Gradient Boosting and Random Forest models.

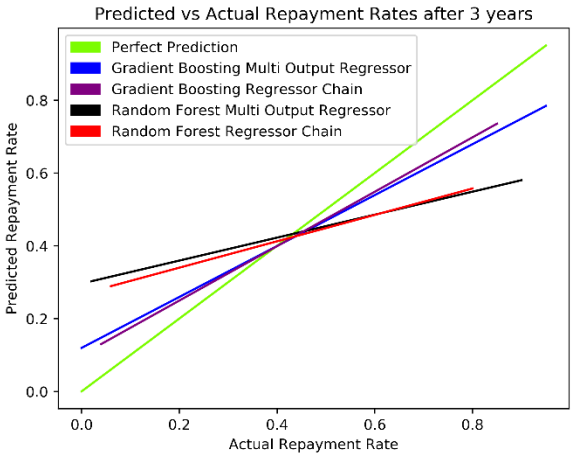


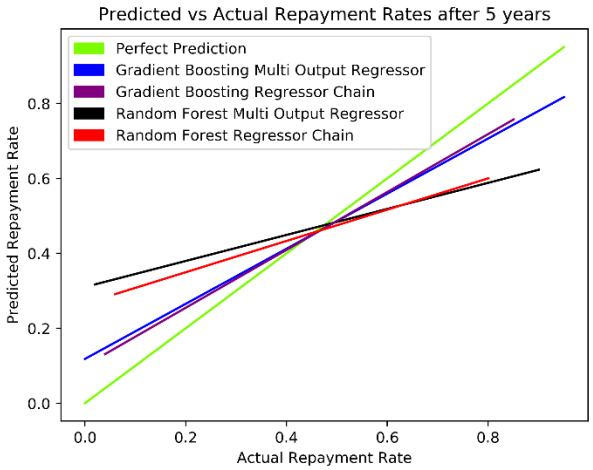


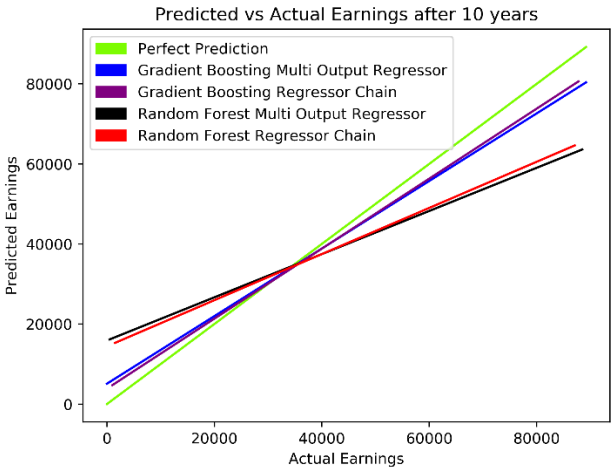


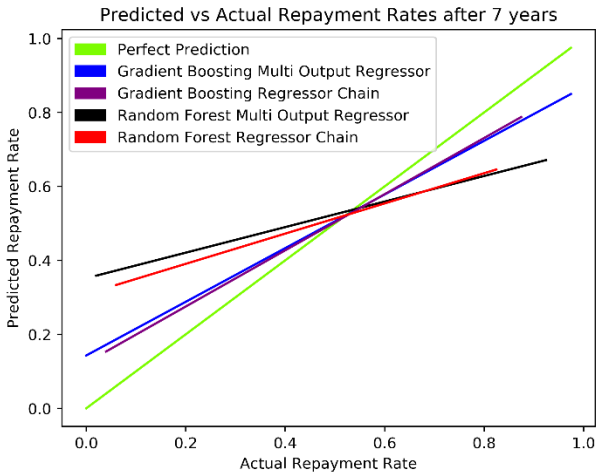
The following images show the results of our models for the 7 target values.

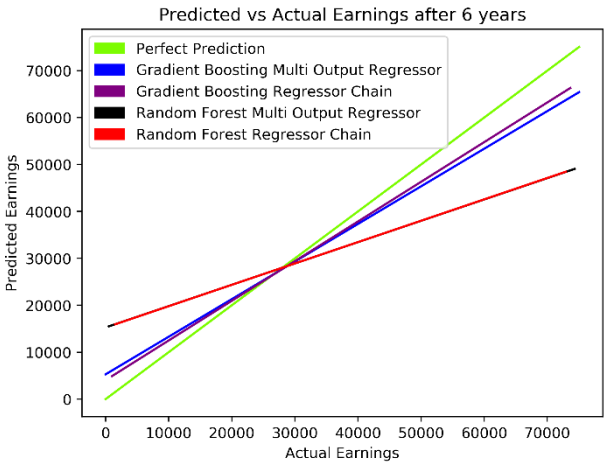


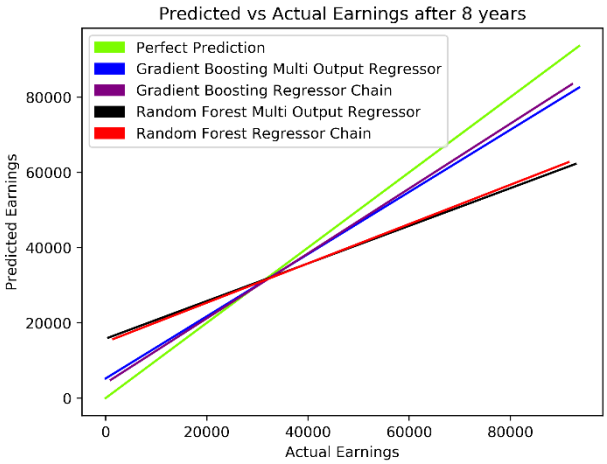



Predicted vs Actual Repayment Rates after 5 years
Perfect Prediction
Gradient Boosting Multi Output Regressor
Gradient Boosting Regressor Chain
Random Forest Multi Output Regressor
Random Forest Regressor Chain
Predicted Repayment Rate
Actual Repayment Rate
0.0
0.2
0.4
0.6
0.8


Predicted vs Actual Earnings after 10 years
Perfect Prediction
Gradient Boosting Multi Output Regressor
Gradient Boosting Regressor Chain
Random Forest Multi Output Regressor
Random Forest Regressor Chain
Predicted Earnings
Actual Earnings
0
20000
40000
60000
80000


Predicted vs Actual Repayment Rates after 7 years
Perfect Prediction
Gradient Boosting Multi Output Regressor
Gradient Boosting Regressor Chain
Random Forest Multi Output Regressor
Random Forest Regressor Chain
Predicted Repayment Rate
Actual Repayment Rate
0.0
0.2
0.4
0.6
0.8
1.0


Predicted vs Actual Earnings after 6 years
Perfect Prediction
Gradient Boosting Multi Output Regressor
Gradient Boosting Regressor Chain
Random Forest Multi Output Regressor
Random Forest Regressor Chain
Predicted Earnings
Actual Earnings
0
10000
20000
30000
40000
50000
60000
70000


Predicted vs Actual Earnings after 8 years
Perfect Prediction
Gradient Boosting Multi Output Regressor
Gradient Boosting Regressor Chain
Random Forest Multi Output Regressor
Random Forest Regressor Chain
Predicted Earnings
Actual Earnings
0
20000
40000
60000
80000