Udacity - Machine Learning Nanodegree Capstone

Predicting the 3-year loan repayment rates in U.S Colleges

By

Saad Khalid

1. Definition

College affordability has become a crucial socio-economic and political issue in the United States in recent history. The presidential elections in 2016 also saw 'making colleges affordable' emerge as a major theme to attract young, millennial voters. However, despite the strong rhetoric across the political spectrum, the newly elected presidential administration has recently revealed its plans to reduce Pell grants (distributed by federal government to students with financial need)[1]. Based on such political developments and the fact that the combined student debt has crossed the 1.2 billion mark[2], it is becoming increasingly pertinent to address the issue with urgency and gain a nuanced understanding of the complex problem.

U.S Department for Education has a dedicated website called 'College Scorecard' that publishes information and data on all colleges and universities across the country. The goal of the website is to help prospective students make informed decisions when choosing colleges or graduate schools. In this project, I used college level characteristics (taken from data published on College Scorecard) to predict debt repayment rates in colleges. More specifically the ***research question*** for the project was:

"Can we accurately predict '3-year debt repayment rates' using college level characteristics in the College scorecard dataset"

As I addressed the central research question stated above, I also identified the key college level characteristics that are predictive of the 3-year debt repayment rates (from now on referred to as repayment rates). Here, it is important to note that my research does not claim to identify causal relationships between college characteristics and debt repayment rates but aims to identify strong associations - characteristics that are important towards predicting debt repayment rates.

My research question presents a prediction problem with a continuous quantitative response (repayment rates lying between 0 and 1). Therefore, I will be using regression methods to predict the outcome. For evaluating the predictive performance of regression models on unseen data, I will be using two metrics.

1. **R-squared on the test set:** R-squared is a measure of goodness of fit. Test set R-squared can be mathematically defined as follows:

   R-squared = 1 – RSS/TSS

   1) RSS = $\sum_i \quad (y_i - y_{\text{pred } i})^2$ 2) TSS = $\sum_i \quad (y_i - \bar{y})^2$

[1] "Bad news for low-income college students in Trump 2017 budget." USA Today. March 21, 2017. Accessed June 13, 2017. http://college.usatoday.com/2017/03/16/bad-news-for-low-income-college-students-in-trump-2017-budget/.

[2] Features, Forbes Special. "How The $1.2 Trillion College Debt Crisis Is Crippling Students, Parents And The Economy." Forbes. August 12, 2013. Accessed June 13, 2017. https://www.forbes.com/sites/specialfeatures/2013/08/07/how-the-college-debt-is-crippling-students-parents-and-the-economy/#6e3a79982e17.

$y_{pred\ i}$ - predictions from the fit model
y_i - actual outcome values in the test set
$\bar{y}$ - mean of the actual outcome values in the test set

2. **Root mean squared error (RMSE) on test data set:** On average how much predictions on the testing data features deviate from the actual values of the outcome in the testing data. Mathematically, it is given as:

$$\textbf{RMSE} = \sqrt{\boldsymbol{RSS}}$$

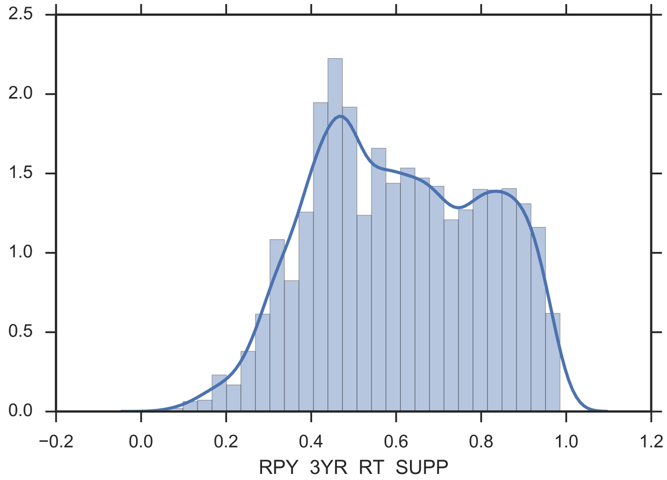
Where RSS is given by the equation 1.

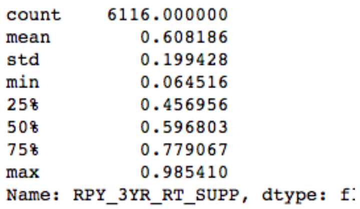
2. Analysis

I used the 'College Scorecard' elements data set to address the aforementioned research problem. It contains data on 122 features on 7703 colleges and universities across the US. The data set also comes with an accompanying data dictionary that describes all of the variables in detail. In order to give the reader an idea of the dataset, here I provide a few examples of the available features. A sample of the dataset is also provided in appendix 1.

Variable name	**Description**	**Type**
CONTROL	Control of institution (whether a college is public, private nonprofit or private for profit)	Categorical (Takes a value of 1, 2 or 3)
HBCU	Flag for Historically Black College and University	Binary/ indicator (takes on value of 1 or 0)
SAT_AVG	Average SAT equivalent score of students admitted	Continuous (Takes a positive value)
PCTPELL	Percentage of undergraduates who receive a Pell Grant	Percentage
MD_EARN_WNE_P10	Median earnings of students working and not enrolled 10 years after entry	Continuous (Takes a positive value)
RET_FT4	First-time, full-time student retention rate at four-year institutions	Continuous (Takes on value between 0 and 1)

The outcome variable of interest in the data set is labeled as 'RPY_3YR_RT_SUPP'. According the data dictionary, the variable is described as 3-year repayment rate of the college/ university. The variable represents a fraction of the student body who is able to repay their college loan/ debts three years after the completion of their program. The underlying distribution of the outcome variable is given below:





```
count    6116.000000
mean        0.608186
std         0.199428
min         0.064516
25%         0.456956
50%         0.596803
75%         0.779067
max         0.985410
Name: RPY_3YR_RT_SUPP, dtype: f
```

Figure 1

The distribution of the outcome of interest seems bimodal upon inspection. We can also see from the summary statistics on the right of the figure that that there are some missing values for output variable. As repayment rates are available for 6116 out of 7703 colleges within, we will have to limit our analysis to the 6116 colleges in the dataset.

To understand the data set, I conducted exploratory data analysis. Here, I present a visualization from process. According to my understanding, distribution of repayment rates in colleges would differ based on whether a college is public, private nonprofit or private for profit (described by the variable CONTROL). Therefore, I tried to verify this assumption:

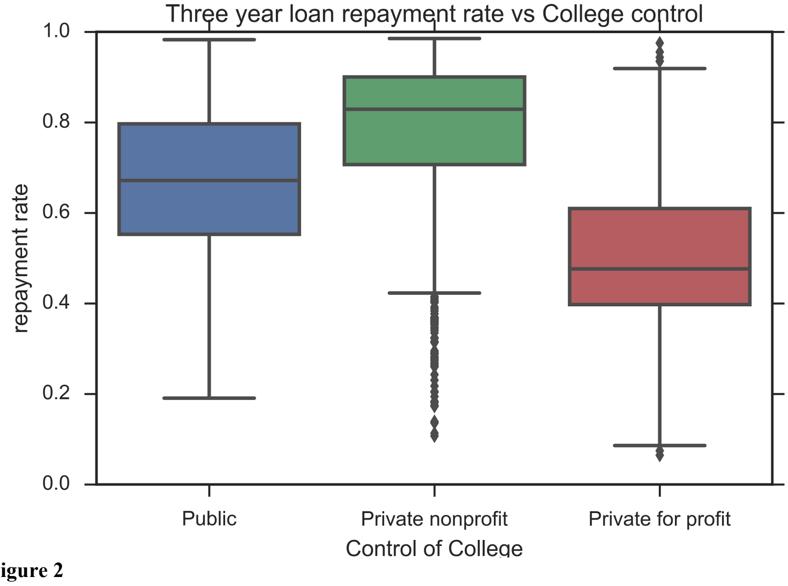


Figure 2

There is a clear variation in the distribution of loan repayment rates across the variable CONTROL. Colleges that are private non profit seem to have the best performance while the private for profit colleges have relatively the lowest repayment rates. The distribution of public colleges lies in the middle. It can also be observed that distribution of repayment rates in private non profit colleges is also narrower as compared to those of the public and private for profit colleges. This visualization conveys that to a certain extent, CONTROL is an informative feature to predict the repayment rates.

2.1. Algorithms and Techniques:

As the outcome of interest, 'college level repayment rate' is continuous, I used regression models to address the research problem. More specifically, I tried out the following techniques:

1. **Linear regression with feature selection:** I started off with simple linear regression. I also employed feature selection techniques before linear regression to choose a certain number of 'best' features from 122 available features (based on 'f-regression' ranking criterion).
2. **Decision tree regression:** Although I initially assumed that Decision Tree Regression would be prone to overfit training data, I decided to see if the optimized model performs better if I tune model parameters such as 'max_depth'.
3. **Lasso and Ridge regression:** I also used regularization techniques such as Lasso and Ridge regression. These techniques also referred to as 'shrinkage methods' shrink regression coefficients to zero (Lasso) or smaller values (Ridge) to avoid overfitting on training data.
4. **Principal components regression(PCR):** Lastly, I tried out PCR which involved reducing the feature space into a few of principal components (linear combinations of features) and using them as features in a linear regression model.

I avoided using higher degree polynomial models because of their tendency to overfit to the training data.

In terms of functionality, the above mentioned algorithms input training data (features and outcomes) and a few algorithm specific parameters.

For training data, preprocessing, feature transformation and feature scaling were important initial steps which will be discussed in the methodology section.

Additionally, a smart choice of the algorithm specific parameters is also crucial in determining the predictive performance of the algorithms (as measured by performance metrics). The table below summarizes the key parameter values I planned to change in order to optimize predictive performance for algorithms.

Model/ Algorithm	Parameters
Linear Regression	Number of 'best' features
Decision Tree Regression	Maximum Depth
Lasso, Ridge Regression	Shrinkage parameter 'alpha'

Principal Components Regression	Number of principal components

2.2. Benchmark:

I set the following benchmark values for performance metrics

Test R-squared: 0.70 (minimum)
Test RMSE: 0.15 (maximum)

Although the benchmark values are hypothetical, they will serve as a standard to compare my results against.

3. Methodology:

This section will discuss the project's methodology. I will start off by describing the steps taken for preprocessing data and making it ready for analysis. Following this, I will also explain the details of implementing and refining the chosen algorithms from the previous section.

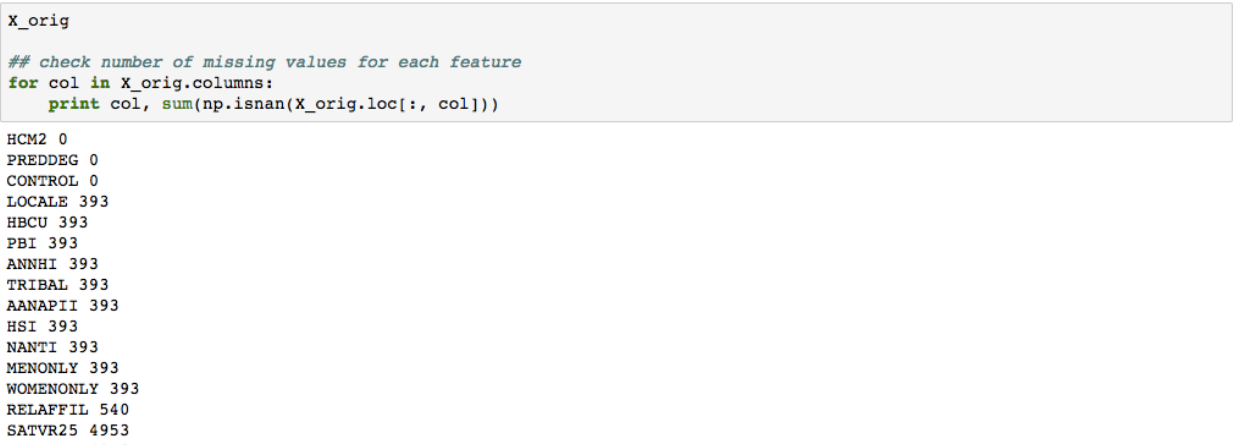
3.1.Data Preprocessing:

Removing ID features: The first step was to simply remove any features that can be used to directly identify a university such as all kinds of IDs, college names, geographical variables, websites etc.

Feature selection: To avoid over fitting on training data, I decided to use the 'SelectKBest' function in sklearn prior to training a linear regression model. I skipped this step for lasso, ridge regression and principle components regression as these algorithms already use techniques such as 'regularization' or 'dimensionality reduction' to reduce the risk of over fitting on training data.

The feature selection step was also important as it also give insight on one of the important components of our research question i.e. determining what factors/ characteristics of US colleges are most predictive of debt repayment rates.

Missing data: As discussed in the previous section, I removed the observations with missing values of debt repayment rates. However, considering the data set had over 100 features, removing the missing values completely from the data set was not a feasible option as doing so would have led to the loss of the entire data set. The following figure shows the number of missing values against each feature:



```
X_orig

## check number of missing values for each feature
for col in X_orig.columns:
    print col, sum(np.isnan(X_orig.loc[:, col]))
```

```
HCM2 0
PREDDEG 0
CONTROL 0
LOCALE 393
HBCU 393
PBI 393
ANNHI 393
TRIBAL 393
AANAPII 393
HSI 393
NANTI 393
MENONLY 393
WOMENONLY 393
RELAFFIL 540
SATVR25 4953
```

To address this, I considered using various types of imputation techniques. First, I thought of using simple 'mean' as an estimate of missing values. But, since a few of the features were categorical and were encoded by whole numbers, the imputed value did not make sense for such features. Therefore, I chose to replace every missing entry with the median of the available feature values. The imputed/ replaced values made sense for features with continuous data as well as categorical data. However, using median to impute missing values also has its limitations as the median may not represent the true value of the missing data.

Feature transformation:

While exploring the data, I found that the distribution of the median income in US university (MD_EARN_WNE_P10 in the data) is highly positively skewed. To satisfy the assumption of linear regression, I transformed the feature by taking the log which made the distribution close to normal. The distributions of the transformed and the original feature can be seen in the following figure.

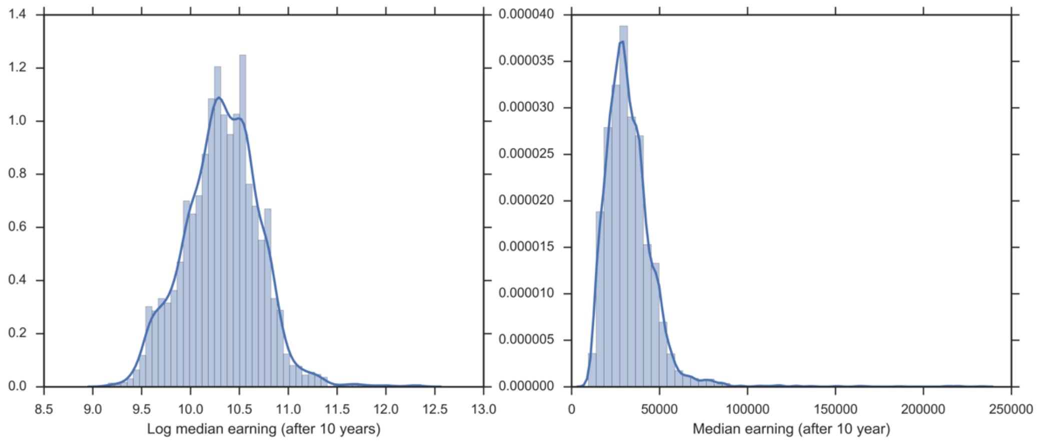


Figure 3

Two variables within the dataset, PREDDEG (predominant degree awarded) and CONTROL are categorical unordered factor variables. The CONTROL variable was explained in the previous section whereas the distribution of the debt repayment rates against the PREDDEG is shown below:

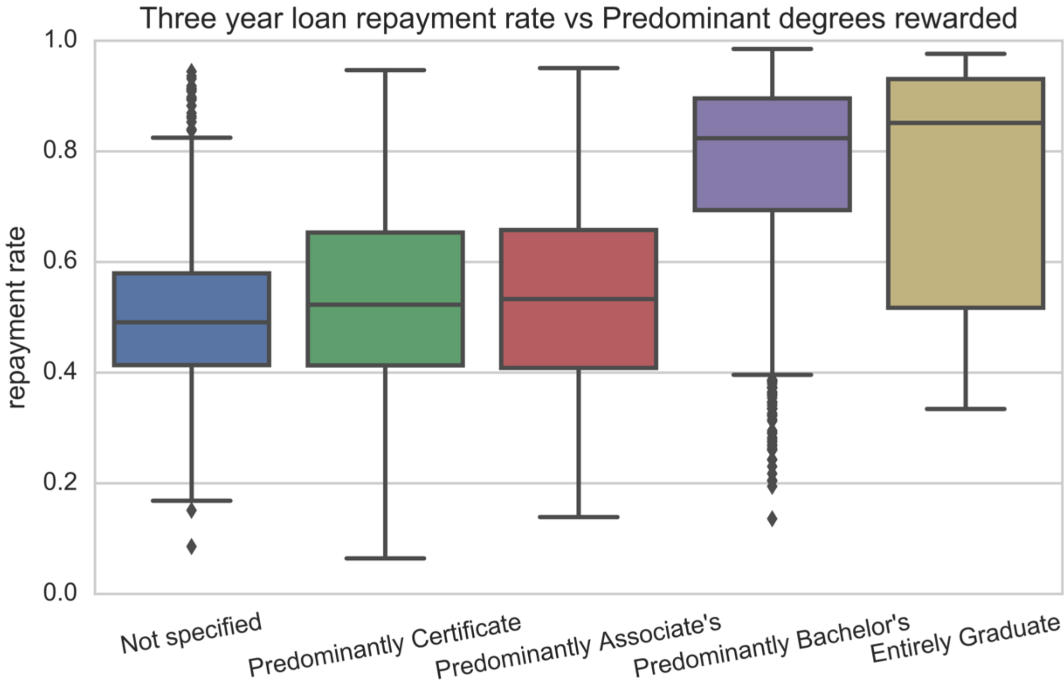


Figure 4

The distributions of repayment rates across the CONTROL and PREDDEG variable (Figure 2 and Figure 4) suggest that it would be better to create dummy variables for improved prediction performance. A single regression coefficient corresponding to the factor variable most likely would not be able to capture the variation across the categories in CONTROL and PREDDEG. Therefore, I used the get_dummies function in 'Pandas' to obtain the dummy variables. Finally, I added the dummy variables in the dataset and removed 'CONTROL' and 'PREDDEG', retaining just the dummies.

I also contemplated the need of feature scaling considering the choice of algorithms. I concluded that since I am using regression methods as predictive algorithms, the units of the features would not affect the results. To verify this, I carried out the analysis with both unscaled and scaled features and compared performance to test my assumptions. I found out that the predictive performance remains the same for all models except for Lasso (further explained in the Conclusion section).

3.2. Implementation:

This section will document the process and code for implementing the chosen algorithms. I will start by discussing the key implementing functions in the accompanying ipython notebook.

1. **impute_missing** function takes in the features matrix as an argument and replaces the missing values based on a method such as 'mean', 'median' or 'mode'. I use this function in the data pre-processing phase to impute the missing values within a variable to the median of the distribution of available values.

2. **kBest_rgr** function takes in k (integer value for number of features) and a regression model as an input. It simply implements a regression model with chosen parameters preceded by feature selection stage where a specified number of features are selected based on the highest 'f-regression' scores. As mentioned in the previous section, I used this function with linear regression models to see if selecting smaller number of features can avoid overfitting on training data and perform better on testing data.
3. **train_regressor** takes in a regression model, training features and training outcome variable as arguments and simply returns a fitted model on the training data.
4. **predict_r2** takes the fitted model, features and the outcomes variables. The function makes prediction (say on test data) using the fitted model and the input features and also evaluates an R-squared score using the computed predictions and input outcome variables. This R-squared is our metric for model evaluation/ performance.
5. **train_predict** combines **train_regressor** and **predict_r2** functions described above in a single function. This function takes in a specified regression model, training features, training outcomes, test features and test outcomes as arguments. It fits a model on the training data and prints the performance metric of the fitted model both on training and testing set. I use this function to test the the performance of all the chosen regression algorithms on the testing data.
6. **fit_model** takes in regression model, range of tunable parameters of the chosen model, features and output variable as arguments. The function uses cross validation to determine the parameters values for the chosen model that give the best predictive performance (on test data) and returns the best model. I use this function abundantly in the 'algorithm refinement' stage. The results of the initial and the 'optimal models' will be presented in the following sub section.

3.3.**Refinement:**

In this section, I will describe the process for improving the chosen algorithms:

1. **Linear regression with feature selection:**

   As mentioned above, initially I used feature selection with linear regression to avoid overfitting on training data. I chose to use 25 best features as inputs to the linear regression as an initial model. To my surprise, adding new features consistently improved performance even on test data. The linear regression model with all 118 features actually performed the best on test data. The performance metrics for a few linear regression models are summarized below:

No. of features	R-squared (test data)	RMSE (test data)
25	0.6619	0.1166
50	0.6925	0.1122
118	0.7235	0.1055

2. **Decision Tree Regression:**

   I started off with the default settings and later evaluated the optimal number of features and value of maximum tree depth through cross validation. The performance of the Decision tree regression model is summarized by the metrics given below:

Parameter values	R-squared (test)	RMSE (test)
max depth = None, features = 118 (Default)	0.5577	0.1332
max depth= 7, features = 86 (Cross validated)	0.6614	0.1158

3. **Lasso and Ridge regression models:**
   I started off with the default parameter setting of Lasso and Ridge regression in sklearn. But later I used cross validation to chose optimal parameter values for the two models (alpha, the degree of regularization). The models evaluated through cross validation also gave me the same performance metrics because they evaluated the alpha value for both Lasso and ridge to be 1.0, which coincidently also happens to be the default value of alpha in the sklearn library. The performance metrics are summarized below:

Model	R-squared (test)	RMSE (test)
Lasso (alpha = 1)	0.3570	0.1608
Ridge (alpha = 1)	0.7324	0.1037

4. **Principal components regression:**
   The principal components regression turned out to perform fairly poorly for the problem at hand. The R-squared for the test set was -231.2121 which suggests that the models performance was way worse than just computing the average of the response variable. The RMSE was also computed to be very high (approximately 3.1). Therefore, I ruled it out immediately as a potential algorithm to address the problem.

4. Results:

Given best performance on test data, I chose Ridge regression with a value of regularization parameter (alpha = 1 and default tolerance = 0.001) as the final model. The performance metrics for the chosen model as mentioned in the previous section are:

Test R-squared: 0.7324
Test RMSE: 0.1037

At this stage, I also checked whether the chosen model is robust by using the random_state parameter in the train_test_split function. I evaluated the model and its performance metrics with 5 different values of random state. The performance metrics for the 5 instances od ridge regression models are given below:

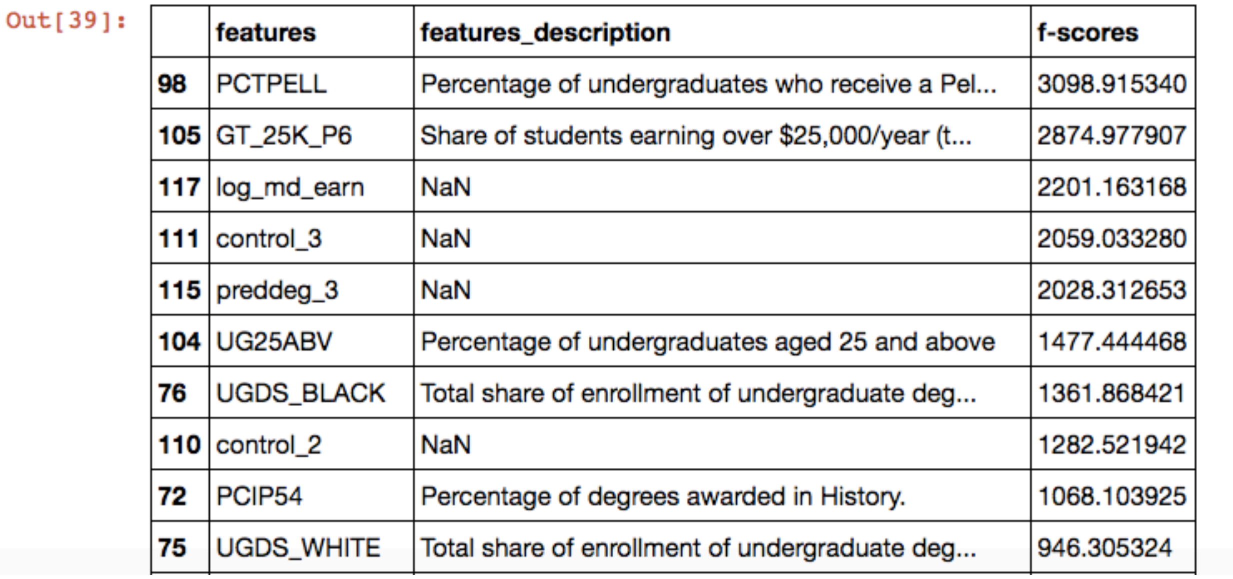
Random state	R-squared (test)	RMSE (test)

1	0.7295	0.1042
2	0.7454	0.0993
3	0.7104	0.1061
4	0.7177	0.1061
5	0.7376	0.1034

From the table above, we can see that the test R-squared and RMSE do not change significantly upon perturbing the input/ training data. Therefore, we can conclude that our choice of model/ algorithm is fairly robust and consistent in its performance on test data.

4.1. Features most predictive of loan repayment rates:

The following table shows the 10 features that are most predictive of the loan repayment rates. The features have been arranged in the decreasing order of their 'f-scores' computed by the SelectKBest function in sklearn library. The second column i.e. 'feature_description', simply contains descriptions of the variable codes in the features column. The features that have NaN in the 'features_ description' column were not part of the original data set and were created at the data preprocessing stage. Hence, the data dictionary does not contain a description of the variables.



Out[39]:

	features	features_description	f-scores
98	PCTPELL	Percentage of undergraduates who receive a Pel...	3098.915340
105	GT_25K_P6	Share of students earning over $25,000/year (t...	2874.977907
117	log_md_earn	NaN	2201.163168
111	control_3	NaN	2059.033280
115	preddeg_3	NaN	2028.312653
104	UG25ABV	Percentage of undergraduates aged 25 and above	1477.444468
76	UGDS_BLACK	Total share of enrollment of undergraduate deg...	1361.868421
110	control_2	NaN	1282.521942
72	PCIP54	Percentage of degrees awarded in History.	1068.103925
75	UGDS_WHITE	Total share of enrollment of undergraduate deg...	946.305324

A few of the above listed features make intuitive sense. It is easy to imagine features that are related to the socio economic status (PCTPELL), earnings (GT_25K_P6, log_md_earn) and race breakdown (UGDS_BLACK and UGDS_WHITE) of the student body to be predictive of the loan repayment of colleges. Similarly, the control of the colleges (dummies control_2, control_3 that code whether the colleges are public, for profit private or not for profit) and predominant degrees awarded (preddeg_3) also are likely to be correlated with loan repayment rates.

However, PCIP54 (Percentage of degrees) is also one of the top ten predictive features. Its correlation with loan repayments is not as obvious. It helps to reminds ourselves that we should think of above listed features features as being correlated, not causally associated with loan repayment rates.

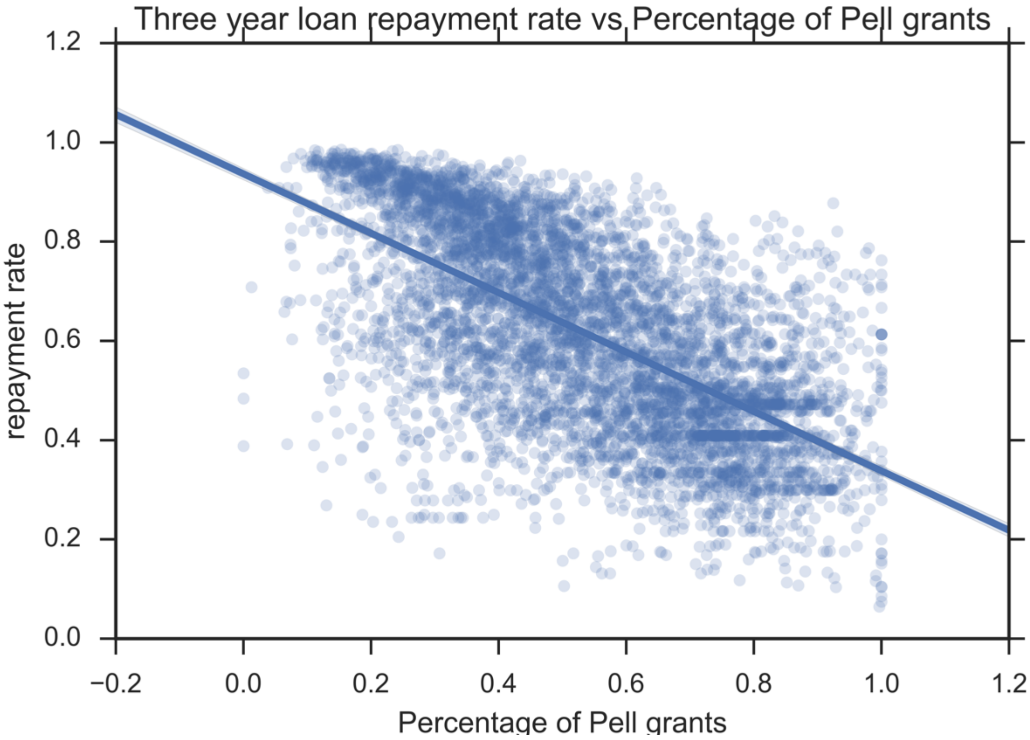
4.2. Comparison to the benchmark:

The benchmark set at the beginning of the study was that the model should explain at least 70 % of the variance in the test data (test R-squared) and should have a maximum RMSE of 0.15. The results from ridge regression have achieved to perform better than the benchmark.

5. Conclusion:

I started the report with a mention of budgetary changes for Pell grants under the new Presidents' administration. In the 'Results' section, I also showed that of all features, the percentage of Pell grant recipients is most predictive of repayment rates. Before moving on to reflections and conclusion for the report, I would like to briefly inspect the relationship between percentage of Pell grants and repayment rates.

5.1. Free form visualization:



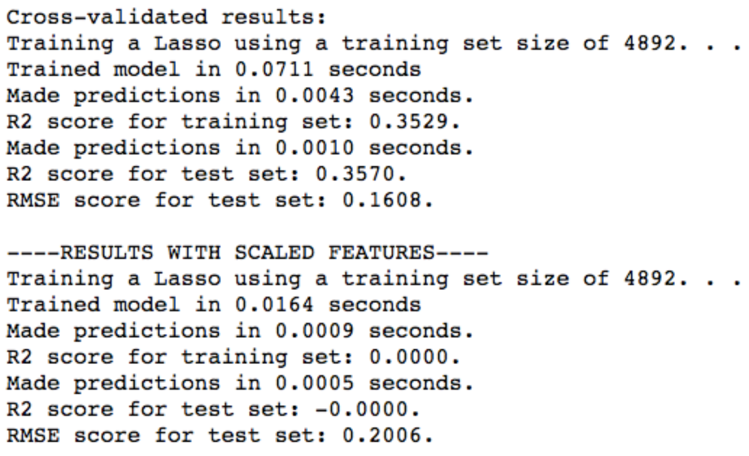
Although not very strong, the figure shows a negative correlation between Percentage of Pell grants and loan repayment rates. This result was a bit surprising to me at first. As a non US citizen, I expected Pell grants to be positively correlated with repayment rates as I imagined the grants to ease loan burden for students. However, later after reading and learning about the nature of Pell grants, I realized that higher percentage of Pell grants must be indicative of a relatively lower socio economic standing of the student body. It is not difficult to see that poorer the student body, lower the college level loan repayment rates.

5.2. Reflection and Improvement:

In this project, through a variety of regression models, I predicted college level loan repayment rates using features in the College scorecard data set. Ridge regression gave the best performance on the testing data based on the two evaluation metrics (R-squared and Root mean squared error). The alpha or shrinkage parameter of Ridge regression was further optimized using cross validation. It turned out that alpha = 1, which coincidentally also is the default value of alpha in sklearn.

A surprising and an interesting part of the project was that for linear regression models, predictive performance on the testing set kept on improving as I added input features. The linear regression model with 118, the maximum number of features gave the best performance. This was contrary to my assumptions that adding too many features will overfit the training data.

Moreover, another interesting aspect was that for most models, feature scaling did not make a difference in predictive performance. This was expected as changes in units of features are compensated by the evaluated coefficients/ weights in regression models. However, for lasso, I got an R-squared of 0 on training data and 0.2 on test data (R^2 = 0.3570 in case of unscaled features) in case of scaled features. This suggests that for scaled features, all the coefficients must have shrunk to zero with just the intercept remaining. I could not explain why lasso showed such a behavior when the features were scaled. A little more reflection and research is required on my part to explain this anomaly.



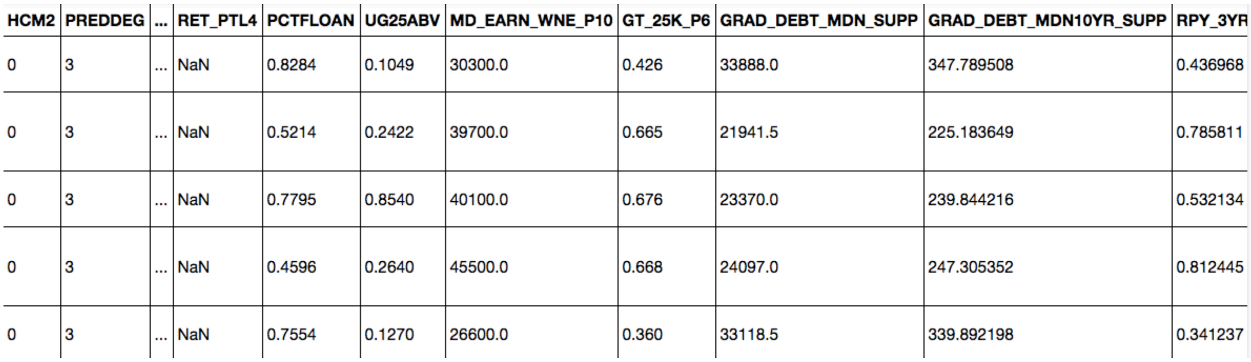
```
Cross-validated results:
Training a Lasso using a training set size of 4892. . .
Trained model in 0.0711 seconds
Made predictions in 0.0043 seconds.
R2 score for training set: 0.3529.
Made predictions in 0.0010 seconds.
R2 score for test set: 0.3570.
RMSE score for test set: 0.1608.

----RESULTS WITH SCALED FEATURES----
Training a Lasso using a training set size of 4892. . .
Trained model in 0.0164 seconds
Made predictions in 0.0009 seconds.
R2 score for training set: 0.0000.
Made predictions in 0.0005 seconds.
R2 score for test set: -0.0000.
RMSE score for test set: 0.2006.
```

There is a lot of missing data in the data set. There could be an improvement in results if advanced imputation techniques were used rather than just replacing the missing values with the median. One of the possibilities could be to predict the missing values using regression or classification models depending on the type of the missing data. However, in my case this would have meant many additional predictive models (118 features in total) to carry out the imputation. Therefore, I took the much simpler way and achieved satisfactory results.

Appendix 1:

Data Set Sample:



HCM2	PREDDEG	...	RET_PTL4	PCTFLOAN	UG25ABV	MD_EARN_WNE_P10	GT_25K_P6	GRAD_DEBT_MDN_SUPP	GRAD_DEBT_MDN10YR_SUPP	RPY_3YR
0	3	...	NaN	0.8284	0.1049	30300.0	0.426	33888.0	347.789508	0.436968
0	3	...	NaN	0.5214	0.2422	39700.0	0.665	21941.5	225.183649	0.785811
0	3	...	NaN	0.7795	0.8540	40100.0	0.676	23370.0	239.844216	0.532134
0	3	...	NaN	0.4596	0.2640	45500.0	0.668	24097.0	247.305352	0.812445
0	3	...	NaN	0.7554	0.1270	26600.0	0.360	33118.5	339.892198	0.341237