Machine Learning Capstone Project

Find university-level factors predict high graduation rate

He Li

2016/07/12

Part 1 Definition

Project overview

Students always want to search for a college that is a good fit for them. Nowadays, public data set can provide students valuable information to help them make their final decision. However, a common problem is, these data sets provides so many features that it is often hard to decide what are factors relevant to one's own question.

In this project, I will use machine learning technique to develop a protocol, so that even a student without any background information can quickly find out factors related to some target variables and make good decisions.

Here I used data downloaded from a public dataset, College Scorecard, https://catalog.data.gov/dataset/college-scorecard. The raw data set, named 'Most+Recent+Cohorts+(All+Data+Elements).csv', contains school information collected before 2015. My task is to select features relevant to student graduation rate, then train a model that can make good predictions based on this simplified data set.

Problem statement

Making predictions of graduation rate is a supervised regression problem.

According to an explanation file of the data set, graduation is defined in two different ways: complete within 150 percent of the expected time to completion, corresponding to features with key word 'C150_' in their names, or complete within 200 percent of expected time, corresponding to features with key word 'C200_'. In this project I will focus on predicting one of these features, C150_4_POOLED, which means pooled completion rates across two years for a 4-year school.

Procedure to solve this problem can be divided into 3 steps.

1. *Separate target variables away from the raw data set. Necessary data cleaning should be applied.*
2. *Feature Selection. Use proper feature selection algorithm to obtain a small sub set of features that are related to C150_4_POOLED.*
3. *Train a linear regression model on the feature subset in step 2. Analyze how these features are related to graduation rate by their weights in final linear model.*

Real meaning of the feature name can be find in data dictionary, CollegeScorecardDataDictionary-09-08-2015.csv, which has also been submitted.

Metrics

In this project I used two metrics to evaluate algorithms.

1. R-square[1]

   R-square indicates the proportion of the variance in the dependent variable that is predictable from the independent variable. Its expression is like:

   $$r^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

   Where SS_{res} refers to residual sum of squares, and SS_{tot} refers to total sum of squares (proportional to the variance of the data). R-square is an often used metric to measure performance of regression model.

2. Running time. Here I only pay attention to running time different feature selection algorithm.

Part 2 Analysis

Data Exploration

At the very beginning, I should split all columns directly related to graduation rate away from raw data set. In later analysis, I will use C150_pooled as label y

of each sample points. However, I noticed that in this data set, graduation information of many samples are lost. So those points should also be removed.

After all these preprocess, there are 2472 sample points with 1688 features in the useful data set.

Part of feature values of a typical data points has been shown below. (It's is not convenient to show all 1688 features)

```
UNITID                                                  100937
OPEID                                                   101200
opeid6                                                    1012
INSTNM                                 Birmingham Southern College
CITY                                                 Birmingham
STABBR                                                       AL
ZIP                                                       35254
AccredAgency      Southern Association of Colleges and Schools
C...
INSTURL                                              www.bsc.edu/
NPCURL                             www.bsc.edu/fp/np-calculator.cfm
HCM2                                                         0
main                                                         1
NUMBRANCH                                                    1
PREDDEG                                                      3
HIGHDEG                                                      3
CONTROL                                                      2
st_fips                                                      1
region                                                       5
LOCALE                                                      12
LATITUDE                                                33.5155
LONGITUDE                                              -86.8536
CCBASIC                                                     21
CCUGPROF                                                    12
```

CCSIZSET	11
HBCU	0
PBI	0
ANNHI	0
TRIBAL	0
AANAPII	0
HSI	0
...	
MD_INC_YR6_N	58
HI_INC_YR6_N	62
DEP_YR6_N	155
IND_YR6_N	PrivacySuppressed
FEMALE_YR6_N	98
MALE_YR6_N	PrivacySuppressed
PELL_YR6_N	87
NOPELL_YR6_N	91
LOAN_YR6_N	168
NOLOAN_YR6_N	10
FIRSTGEN_YR6_N	34
NOT1STGEN_YR6_N	134
OVERALL_YR8_N	207
LO_INC_YR8_N	PrivacySuppressed
MD_INC_YR8_N	63
HI_INC_YR8_N	100
DEP_YR8_N	197
IND_YR8_N	PrivacySuppressed
FEMALE_YR8_N	108
MALE_YR8_N	PrivacySuppressed
PELL_YR8_N	72
NOPELL_YR8_N	135

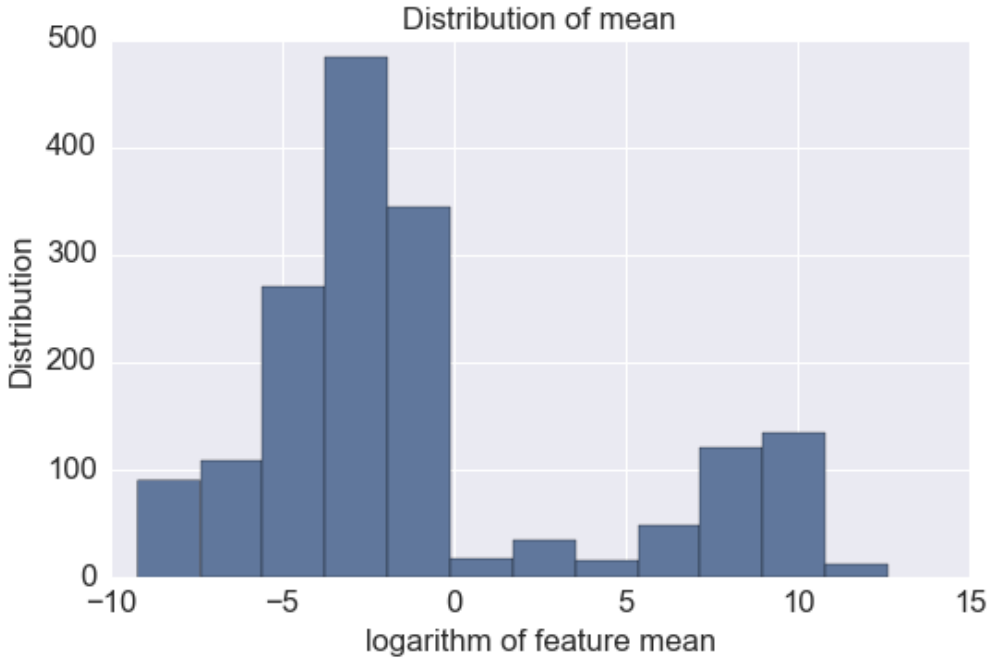
```
LOAN_YR8_N                                    PrivacySuppressed
NOLOAN_YR8_N                                  PrivacySuppressed
FIRSTGEN_YR8_N                                PrivacySuppressed
NOT1STGEN_YR8_N                                             165
REPAY_DT_MDN                                             379852
SEPAR_DT_MDN                                             379852
REPAY_DT_N                                                  344
SEPAR_DT_N                                                  371
```

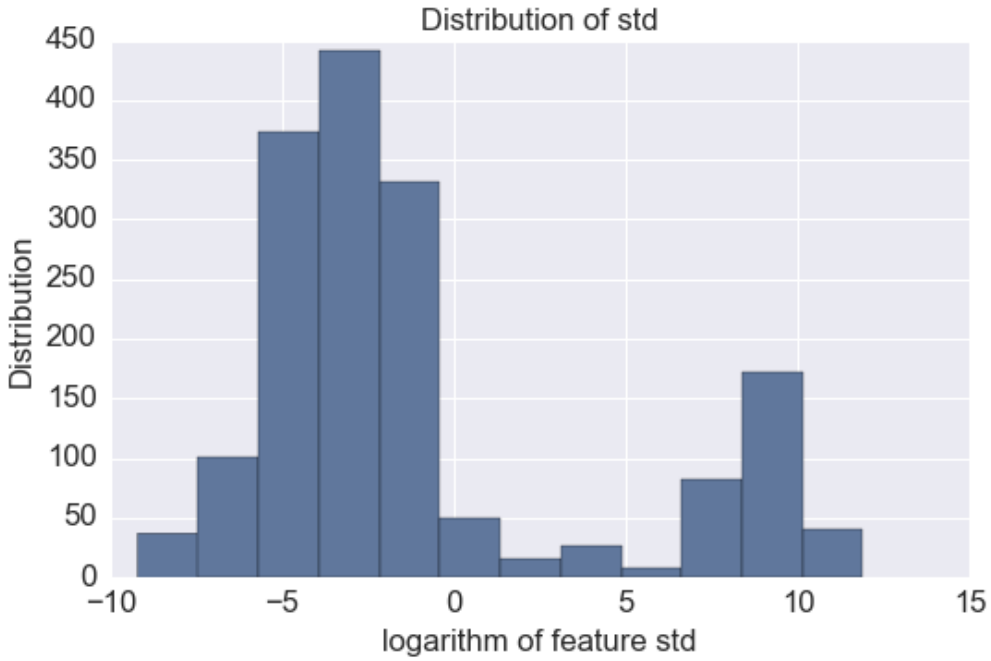
One could notice 3 points:

1. The first 10 features are just basic school information, such as school name, ID, location, etc. They are obviously irrelevant features.
2. Numeric values of various features can be quite different. This remind me may be feature standardization is necessary.
3. Several elements are protected for privacy purposes. This missing values should be replaced.

Exploratory Visualization

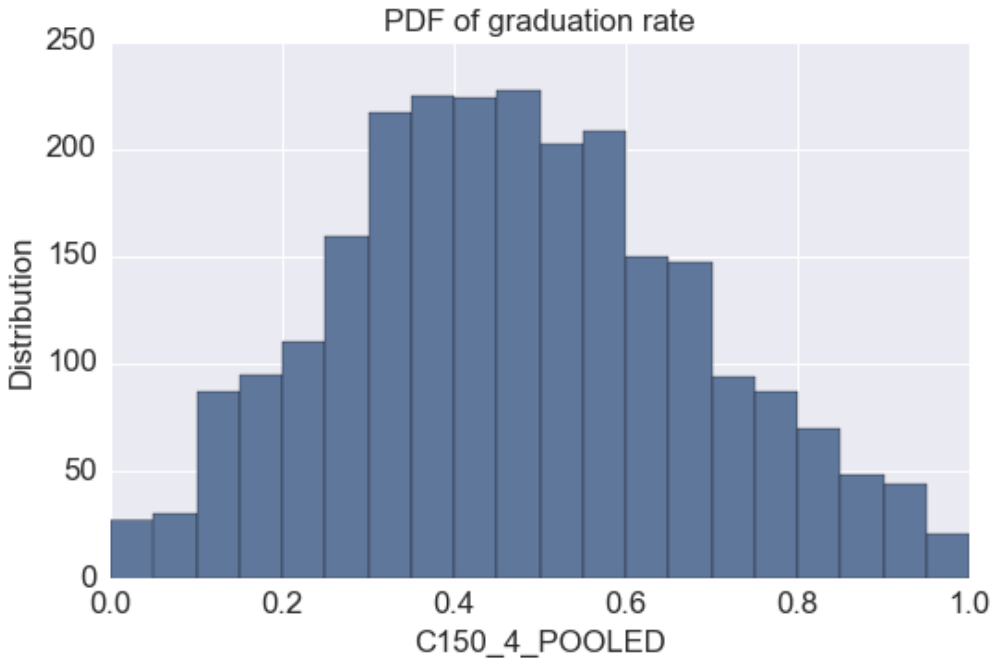
In first part of this section, I have shown the distributions of mean value and standard deviation of all features. The range of these two statistics turn out to be quite wide. So I have to plot the x axis in logarithm. I have imported python package ‘seaborn’ when plotting the figures.





From the two figures above, I'm convinced range of feature values are so different that I need to use feature standardization before training a linear model.

Secondly I took a look at the distribution of label variable, C150_4_POOLED.



It looks like a normal distribution. Here I calculated outlier step as 1.5 times the interquartile range (IQR), and removed those data points whose label value is beyond an outlier step of the IQR. This treatment is meant for eliminating the influence of outlier points with two high or too low graduation rates.

Algorithms and Techniques

Main techniques used in this project are feature selection Algorithms and linear regression model.

Feature selection

Here's a list of feature selection algorithms:

1. *Removing features with low variance*
2. *Univariate feature selection*
3. *Recursive feature elimination*
4. *Lasso.*

Method 1 is invalid because just as I mentioned in last section, features will be standardized by their mean and standard deviation.

Method 2, Univariate feature selection works by selecting the best features based on univariate statistical tests. This algorithm will run an F-test [2] for each feature and the target variable, and return a list of features with high cross correlation with target variable.

Advantage: It is simple to use and fast.

Disadvantage: It can only select features that are linearly correlated with dependent variable. And this method doesn't take the model into account, so that the feature set it returns may not be the best one for a specific model.

Method 3. Given an external estimator that assigns weights to features (e.g., the coefficients of a linear model), recursive feature elimination (RFE) is to select features by recursively considering smaller and smaller sets of features. First, the estimator is trained on the initial set of features and weights are assigned to each one of them. Then, features whose absolute weights are the smallest are pruned from the current set features. That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached.

Advantage: This algorithm choose features directly by their performance on the model. Thus it more likely to return a feature set that are suitable for the model.

Disadvantage: It is slow, and needs too much calculation.

Method 4. Lasso is short for 'Least Absolute Shrinkage and Selection Operator'. It is a Linear Model trained with L1 prior as regularizer. It is a beautiful regression model, first raised by Professor Tibshirani, 1996[3]. Its expression is quite similar to ridge regression, but use L1 norm of weights vector in the regularization term. This slight change will make the final weights vector rather sparse, which means weights on irrelevant features will be set to zero.

Advantage: When using Lasso, 'feature selection' and 'model training' can be done at the same time. It is relatively fast and accurate.

Disadvantage: Optimization of cost function with L1 norm regularization term is hard for calculation.

regression

Once the relevant features are decided, I will train a linear regression model [4].

Linear regression model is the simplest regression model. It might not defeat many non-linear regression model in accuracy, for example SVR with non-linear kernel, or decision tree repressor. But it is the most proper algorithm for this task.

Advantage of linear regression is that its result is easy to explain. Features with large positive weights indicate they have highly positive correlation with completion rate, where large negative weights imply high negative correlation.

To implement these algorithms, I used feature selection module [5] and linear_model module [6] in sklearn package. When performing univariate feature selection with function SelectKBest, I set score_function = f_regression. When using Lasso, I set max_iter = 5000 to guarantee convergence of Lasso regressor. Other parameter will be tuned in practice.

I have also imported time module to calculate running time of each algorithm.

Benchmark

In this section, I trained a linear regression model with L2 norm regularization term, called ridge regression, on the entire data set before feature selection. Since # of sample points is 2472, but # of numerical feature is 1678, I believe there could exists a horrible overfitting problem. To fix it, I chose a large regularization factor Alpha, up to 100.

I used train_test_split function provided by sklearn package, to randomly select 75% of sample points as training set and remaining 25% as test set. Performance of each model in this project will be measured by its r-square score on this test set.

For ridge regression [7] trained on entire data set:

R-square on training set: 0.9208

R-square on test set: 0.6384

I found that overfitting still exists, for r-square score is much lower on test set even when regularization factor alpha = 100. What's more, ridge regression can't tell us which feature is necessary for predicting independent variable.

I expect feature selection can will fix the overfitting problem.

Part 3 Methodology

Data Preprocessing

I have fixed 3 main problems in raw data set in this ‘Data Preprocessing’ section.

1. *Data type of several numerical values are 'string'. Change them into 'float' type.*
2. *Replace missing feature values with average of corresponding feature.*
3. *Standardize features into zero mean and unity variance.*

I wrote a function 'FixData_manually' to implement step 1 and step 2. As to step3, I directly use function from sklearn.preprocessing module.

Implementation

In this section, I compared 3 feature selection algorithm mentioned in 'Algorithms and Techniques'. In practice, I first selected 30 features by each method, then trained a ridge regression model with regularization parameter alpha = 10 on the selected feature set. These methods are evaluated by time spending on feature selecting process and how well the chosen features do in making predictions.

1. **Univariate feature selection**

   Function used: SelectKBest.

   Parameter setting : scoring_func = f_regression, and k = 30.

   Running time: 0.083s

   R-square on training set: 0.6656

   R-square on test set: 0.6129

2. **Recursive feature elimination**

   Function used: RFE.

   Parameter setting: estimator = linear_model.Ridge(alpha = 10), n_feature_to_select = 30, step = 1

   Running time: 344.3s

   R-square on training set: 0.7405

   R-square on test set: 0.6906

3. **Lasso**

   Usage of Lasso is quite different from the other two algorithms. It can't set number of selected features in advance. So this time I manually tuned the regularization parameter alpha in order to make the number of non-zero feature weights = 30.

   Function used: linear_model.lasso

Parameter setting: alpha = 0.021, max_iter = 2000

Running time: 0.6362s

R-square on training set: 0.7308

R-square on test set: 0.6945

Make a summary on above results.

Univariate feature selection is the fastest algorithm in these three, but prediction accuracy is more or less unsatisfactory. Recursive feature elimination's accuracy is better than univariate feature selection. However it takes too much time dealing with huge data set. Lasso turns out to give us the highest r2 score and also much faster than RFE. So I finally decided to use lasso in feature selection process.

Refinement

There are two parameters I need to tune in this section, the number of remaining features and regularization parameter in ridge regression model. However, this is not a 'the more accurate, the better' game. It is much more important to capture features that dominant graduation rate prediction, than a slightly improvement in r2 score.

Here I require the size of selected feature set should be smaller than 30, and the final r-square score on test set larger than 0.69.

Sklearn provides a useful tool LassoCV to select the best model for Lasso by cross validation. But in practice it turns out to be too slow dealing with over 1600 features. **I came up with a trick when working out this section: use a combination of Univariate feature selection and LassoCV.**

The idea is like this:

Step 1. Apply univariate feature selection to quickly delete features that are apparently irrelevant to target variable. Reduce feature number to 500 after this step.

Step 2. Run LassoCV on this much smaller data set.

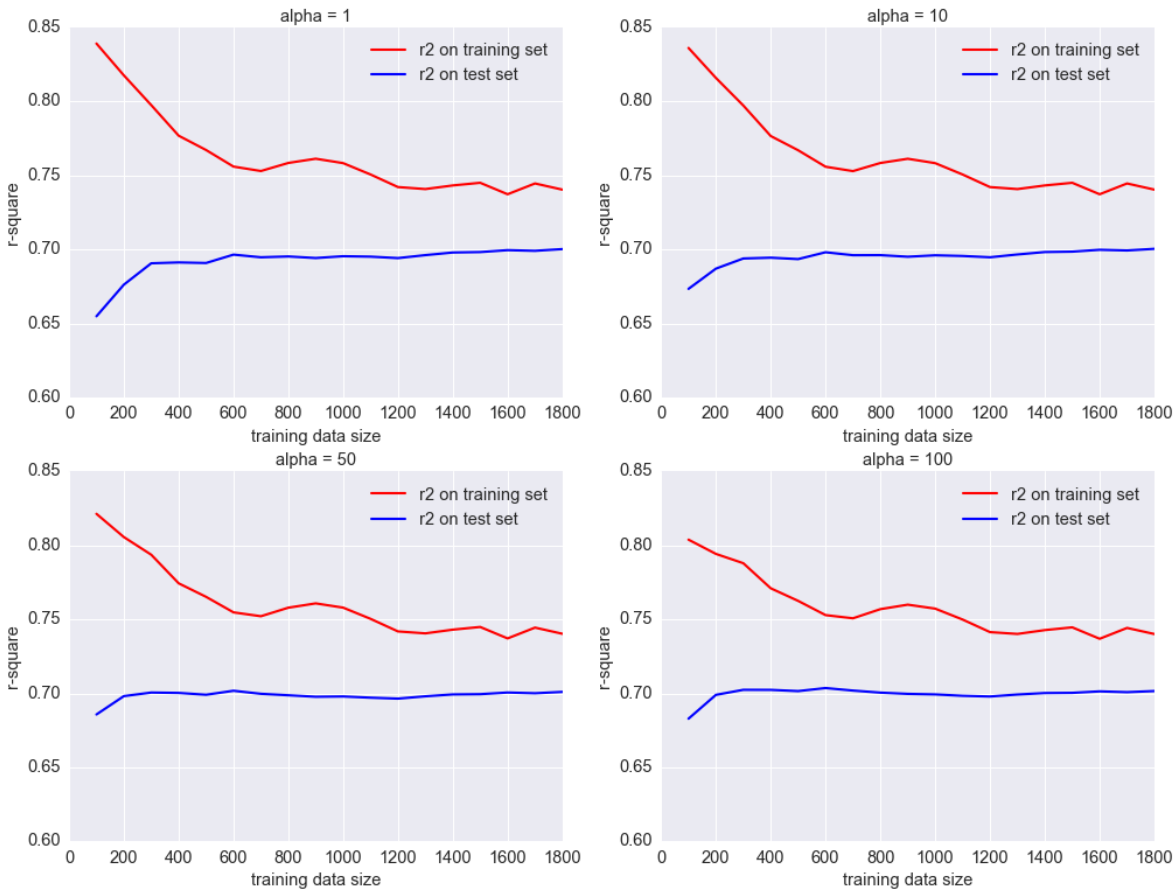
I tuned threshold for LassoCV in a threshold list [2*mean, 4*mean, 6*mean, 8*mean, 10*mean], manually. After feature selection, I trained a ridge regression model, with regularization parameter automatically optimized by GridSearchCV.

Finally I figure out when threshold = 8*mean, alpha = 100, feature number reduced to 20 but r2 score on test set increase to 0.7013 at the same time. This final solution is better than solution I found in section 'Implementation'.

Part 4 Results

Model Evaluation and Validation

In this section, I have shown the learning curve for various training set size, and looked at how the learning curve would change for different regularization parameter alpha.



One can see that for various alpha, r-square on test set becomes quite stable when training set size is larger than 600. This fact indicates overfitting is well prevented and the final model is robust for changes in training set.

Justification

Comparing the final model and benchmark reported earlier, one could notice improvement is great. There is serious overfitting problem in benchmark model. And it cannot tell which feature is necessary in making prediction. The final model has reduced feature number to 20. And it has higher r2 score on test set

than benchmark result. Based on the weights of features in final model, one can easily tell what factor is important to predict retention rate of a specific university. Detailed analysis on these factors will be provided in section 'Free-Form Visualization.

Part 5 Conclusion

Free-Form Visualization

Finally, based on the final model, I could answer the question raised in Part 1.

Here are the factors selected by algorithm and their weights in a linear regression model.

```
TUITIONFEE_OUT                0.022693
RET_FT4                      0.022461
COMP_ORIG_YR3_RT              0.022369
PFTFTUG1_EF                  0.021209
NOT1STGEN_COMP_ORIG_YR8_RT     0.021049
SAT_AVG                      0.019191
NOT1STGEN_WDRAW_ORIG_YR8_RT    0.017462
COMP_ORIG_YR4_RT              0.015916
D200_4                       0.015593
COMP_ORIG_YR6_RT              0.014750
DEP_COMP_ORIG_YR3_RT          0.013581
main                         0.013385
CUML_DEBT_P10                0.013150
HI_INC_COMP_ORIG_YR4_RT        0.007551
WDRAW_4YR_TRANS_YR6_RT        -0.012327
IND_COMP_2YR_TRANS_YR6_RT     -0.013335
FEMALE_WDRAW_4YR_TRANS_YR4_RT  -0.014035
```

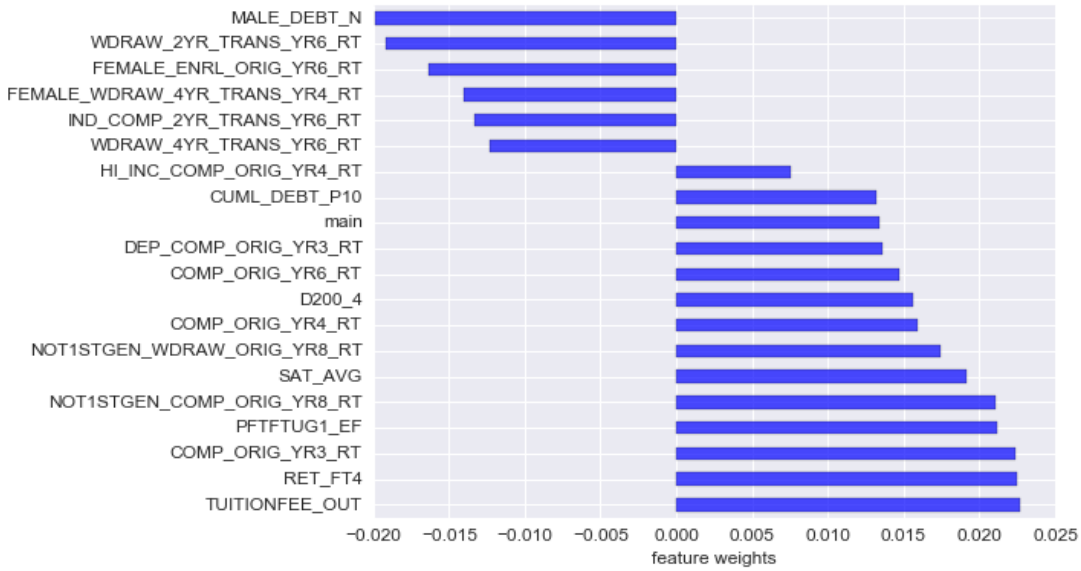
```
FEMALE_ENRL_ORIG_YR6_RT        -0.016311

WDRAW_2YR_TRANS_YR6_RT         -0.019129

MALE_DEBT_N                    -0.019892
```

How important each feature is and whether it is positively or negatively correlated to graduation rate is clearly shown in the bar plot below, where y-axis is feature name, x-axis refers to corresponding weights. Their meaning can be found in the Data Dictionary.

.



Reflection

In summary, to solve this problem, I first replaced missing data points with the average value of each columns. Secondly, I have tried three different feature selection technique and decided to use a combination of Univariate feature selection and Lasso regression. After carefully selecting process, I simplified huge data set with over 1000 features into a small data set with only 20 features relevant to graduation rate prediction. Finally, I trained a ridge regression model on the simplified training set, which can not only predict graduation rate accurately, but also presents readers how these selected features are correlated with the target variable.

In my opinion, Lasso is really a charming method. It is a smart idea to change the complicated feature selection problem into a convex optimization problem, which mathematicians are already familiar with.

Improvement

In this project, I selected features only based on their performance on linear models. It is possible that some features don't affect the dependent variable in a linear way, but I neglected these possibilities. I wonder if there are any feature selection algorithms that are able to distinguish feature non-linearly correlated to dependent variable.

Reference

[1] https://en.wikipedia.org/wiki/Coefficient_of_determination

[2]To know more about correlation analysis in statistics, https://en.wikipedia.org/wiki/Correlation_and_dependence

[3] Robert Tibshirani, **Journal of the Royal Statistical Society. Series B (Methodological)** Vol. 58, No. 1 (1996), pp. 267-288

[4] Linear regression: https://en.wikipedia.org/wiki/Linear_regression

[5] Online documents for feature selection: http://scikit-learn.org/stable/modules/feature_selection.html#feature-selection

[6] Online documents for linear model: http://scikit-learn.org/stable/modules/linear_model.html#

[7] To learn more about regularization used in regression: https://en.wikipedia.org/wiki/Tikhonov_regularization