Ranking Prediction of Higher Education Institutes using Financial and Expenditure Data

Mohamad Zeini Jahromi
Sep 18th, 2017

Project Overview

The total number of awards, certificates, and degrees completed each year in post-secondary education institutions have been widely used to evaluate their performances and are an indication of relative success and ranking of these institutions throughout the nation. Many studies have investigated the effects of different parameters (such as financial aid, institutes funds, revenue, expenditures and etc.) on the institutes completion rates. The results of these studies could help institutions to decide how to allocate funds to their segments more effectively and create a well-balanced money flow within their systems. On the other hand, the costs of higher education in the United Sates are high and it's being considered as an investment and consequently having a knowledge of success rate and ranking of a specific institution would help both students and their families in making the right decision.

This project focuses on studying the relationship between institutions financial aid and expenditure and their respective completion rates (the total number of awards, certificates, and degrees completed). Furthermore, a predictive model will be developed in order to predict the completion rates using the financial aid and expenditure data of institutions.

This study is inspired by Udacity's capstone project guidelines and uses the IPEDS dataset (Integrated Postsecondary Education Data System Delta Cost Project Database), which is based on surveys on finance, enrollment, staffing, completions, and student aid of post-secondary education institutions within the US.

Problem Statement

As stated above, the completion rates (the total number of awards, certificates, and degrees completed) is a measure of success and ranking of institutions. This study is seeking to find out whether the completion rates could be predicted based on financial aid and expenditure data from post-secondary education institutions and in the following, the results will be analyzed to see which parameters are more significant in terms of predicting the target label (completion rates).

Solution Strategy

Different algorithms (*K-Nearest Neighbor Regressors*, *Support Vector Regressors*, *Random Forest Regressors* and etc.) will be used to create predictive models. Eventually, these models will be able to predict a specific institution completion rate based on its expenditure or financial aid features.

The data cleaning process will be performed on the dataset to eliminate outliers or missing values and then cross-validation is used to split the dataset into a training and a testing datasets.

The Principal Component Analysis will be implemented to find the dimensions that explain the most of the variance. The top components become predicting features for creating the model.

Before the PCA procedure, the standardizing and transforming procedures will be performed on the dataset. The models will be tuned using Grid Search function using sets of hyperparameters.

Evaluation Metrics

It is difficult to measure the quality of a given model without quantifying its performance over training and testing. This is typically done using some type of performance metric, whether it is through calculating some type of error, the goodness of fit, or some other useful measurement. For this project, we will be calculating the coefficient of determination (R^2) to quantify our model's performance. The coefficient of determination for a model is a useful statistic in regression analysis, as it often describes how "good" that model is at making predictions.

The values for R^2 range from 0 to 1, which captures the percentage of squared correlation between the predicted and actual values of the target variable. A model with an R^2 of 0 is no better than a model that always predicts the mean of the target variable, whereas a model with an R^2 of 1 perfectly predicts the target variable. Any value between 0 and 1 indicates what percentage of the target variable, using this model, can be explained by the features. R^2, however, inflates when adding more predictors (variables) to the metric. The variance caused by predicted value hence increase even without model improvement. Adjusted R^2 score is developed to counter the inflation and adding a penalty for additional variables entering the metric. Adjusted R^2 is always smaller or equal to R^2 score. A model can be given a negative R^2 as well, which indicates that the model is arbitrarily worse than one that always predicts the mean of the target variable.

Data Exploration

The dataset used in this project are from IPDES database (Integrated Postsecondary Education Data System Delta Cost Project Database), which is based on surveys on finance, enrollment, staffing, completions, and student aid of post-secondary education institutions in the US. It consists of many features including financial aid, revenue, and expenditure of institutes across the US. The datasets for academic years of 1987-1999 and 2000-2012 have the total number 215613 entries and 974 features. All expenditure categories (20 features) are used as the **Expenditure Features** and in the same way, the financial aid categories (5 features) are used as the **Financial aid Features**. Also, the categories that contain total completions value are going through further preprocessing to create the **Completion Rates Label**.

The selected features are numerical, and since the dataset contains a significant amount of null value (missing data or unreported data), multiple data cleaning scenarios will be performed which results in eliminating some features and data points.

Feature Observation

The following table (**Table 1**) shows the data subset features that are being selected to be used in this project. The table has 27 rows including 20 Expenditure features, 5 Financial aid features, the **totalcompletions** target label and the **has_completion** tag. A brief explanation of each feature is provided in the next column.

Table 1- The data subset features and label

Eligible data points	
has_completion	Indicator of whether totalcompletions is reported.
Label	
totalcompletions	The total number of degrees, awards, certificates granted of the current year.
Features (Expenditure)	
instruction01	Instructional expenses for the institution and excludes administration, operations and maintenance.
research01	Expense used to produce research outcomes excluding operation and maintenance, interest amounts attributed to the research functions.
pubserv01	Expense category that provides noninstructional services beneficial to individuals and groups external to the institution such as conferences.
acadsupp01	Expenses to support instruction, research, and public service. This category includes retention, preservation, and display of education materials.
studserv01	Expenses associate with admissions, registrar activities, and activities that contribute to students' emotional and physical well-being and their intellectual, cultural, and social development outside the formal instructional program.
instsupp01	Expense for day-to-day operational support of the institution such as space management, employee personnel, and records.
acadinststud01	Academic and institutional support and student service total of current year
opermain01	Expenses for operations providing service and maintenance related to campus grounds and facilities. Institutions may optionally distribute depreciation expense to this function.
depreciation01	Total depreciation of current year
grants01	The sum of all operating expenses associated with scholarships and fellowships including payment in support of the cost of education or third parties for off-campus housing.
auxiliary01	Expense of all operating associated with essentially self-supporting operations of the institution such as student health services. The amount of interest is excluded.
hospital01	Operating expenses related to a hospital run by the postsecondary institute.
independ01	Expenses associated with operations that are independent of or unrelated to the primary missions of the institution.
totaloper01	Total expenses is the sum of all operating expenses that result from providing goods and services.
otheroper01	All expense other than categories above which discontinued after the Academy year 2010.
othernon01	All other non-operating expense of current year
other01	All other expense
eandg01	Total education and general expenditures includes all core operating expenditures, including sponsored research, but excluding auxiliary enterprises.
rschpub01	Expense for research and public service of current year
acadinststud01	Expenditures for academic and institutional support and student services.
Features (Financial aid):	
appliedaid01	Discounts and allowances applied to tuition and fees are reductions to the amount charged for tuition and fees by the application of scholarships and fellowships.
grant07	The sum of Pell, Federal, State, Local, and Institutional Grants (funded and unfunded).
tuition_discount	That part of a scholarship or fellowship that is used to pay institutional charges such as tuition and fees or room and board charges.
any_aid_num	Number of full-time, first-time degree/certificate-seeking undergraduate students who received any financial aid including grants, loans, assistantships, scholarships, fellowships, tuition waivers, tuition discounts, veteran's benefits, employer aid (tuition reimbursement) and other monies (other than from relatives/friends) provided to students to meet expenses.
loan_num	Number of full-time, first-time degree/certificate-seeking undergraduate students who received student loans.

Data Cleaning

Before data can be used as input for machine learning algorithms, it often must be cleaned, formatted, and restructured (preprocessing). Some initial observations are as follows:

- The **has_completion** feature shows that only 142220 out of total number of 215613 entries are eligible institutions with the completion rates data.
- 166367 of data points have negative values (which does not make sense in terms of dollars)
- 3766731 of data points have Null value or missing.

In the first step, all the entries without the completion rates data are removed. Next, the features that include more than 80% of Null or missing values are removed. Then, the remaining data points that have Null or missing values, or negative values are eliminated.

After initial data cleaning procedure, the dataset is left with 4609 entries and 21 columns (including 19 features, the **totalcompletions** target label, and the **has_completion** tag). This preprocessing significantly improves the accuracy of outcomes and the predictive power of learning algorithms.

Calculate Statistics

In the following, **Table 2** summarizes the descriptive statistics of Expenditure/Financial datasets are calculated. *numpy* library has already been imported and used to perform the necessary calculations. These statistics will be extremely important later on to analyze various prediction results from the constructed model. The minimum, maximum, mean, median, and standard deviation of few features are presented in the table.

Table 2- Descriptive statistics of expindature and financial aid datasets

	Total completions	instruction01	pubserv01	acadsupp01	opermain01	grants01	any_aid_num	loan_num
count	4609	4.61E+03	4.61E+03	4.61E+03	4.61E+03	4.61E+03	4609	4609
mean	2153.59	6.68E+07	1.19E+07	1.61E+07	1.48E+07	1.16E+07	1098.76	572.96
std	3137.57	1.39E+08	3.59E+07	3.22E+07	3.06E+07	2.27E+07	1508.39	751.63
min	14	5.38E+03	8.30E+01	1.03E+03	1.33E+03	7.80E+01	6	0
25%	529	1.17E+07	3.96E+05	2.22E+06	2.80E+06	1.91E+06	301	81
50%	1100	2.47E+07	1.61E+06	5.22E+06	6.11E+06	4.67E+06	638	300
75%	2555	6.11E+07	5.92E+06	1.46E+07	1.43E+07	1.15E+07	1345	815
max	43561	2.52E+09	4.05E+08	4.05E+08	6.52E+08	3.52E+08	23964	9632

Outlier Detection

Detecting outliers in the data is extremely important in the data preprocessing step of any analysis. The presence of outliers can often skew results which take into consideration these data points. There are many "rules of thumb" for what constitutes an outlier in a dataset. Here, Tukey's Method is used for identifying outliers: An outlier step is calculated as 1.5 times the interquartile range (IQR). A data point with a feature that is beyond an outlier step outside of the IQR for that feature is considered abnormal. Once this implementation has performed, 67 entries are detected as outliers and removed from the dataset.

Exploratory Visualization

To get a better understanding of the dataset, both heatmap and scatter matrix of all the features are constructed and illustrated in **Figure 1** and **Figure 2**.

There are high correlations between most of the variables. For instance, **instruction01** feature correlates with most of the variables specially with **eandg01, totaloper0** and **opermain01** features. Few features like **tuition_discount** and **other01** have no significant correlation with other features. As the plot shows, expenditure feature sets are highly correlated with almost all of financial aid features.

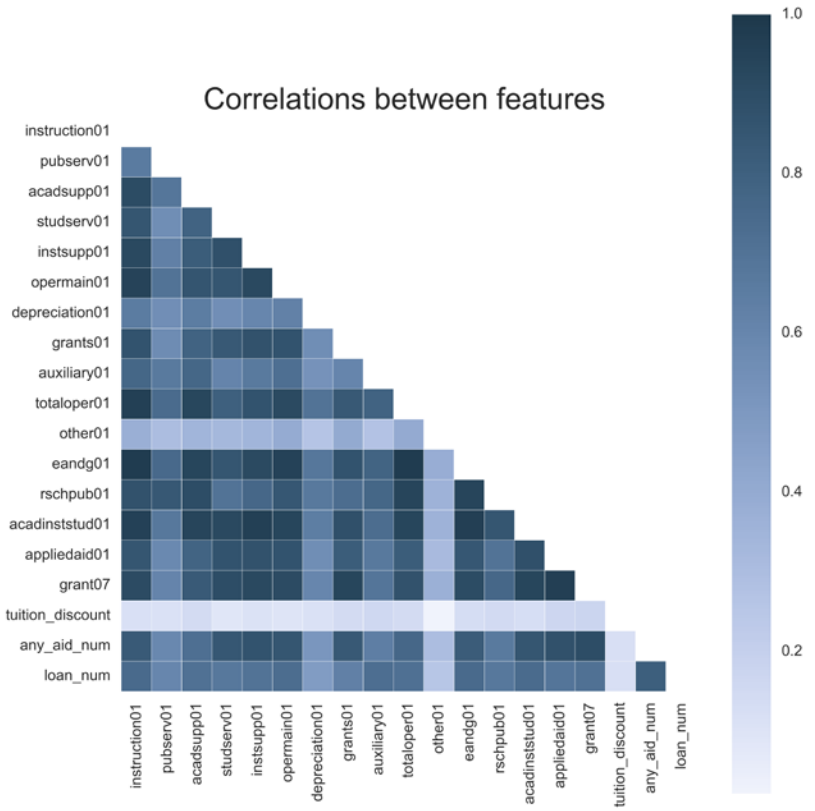

Figure 1- Heatmap of features showing the correlations between features.

Feature Distributions

Scatter matrix is able to demonstrate whether the feature we attempted to predict are relevant to each other or not. It also, revels if the distribution is normal or skewed. Considering **Figure 2**, the relatively large standard deviations indicate the distributions are spread out. In addition, it shows the distributions are highly skewed for almost all the features. The feature rescaling and data transformation could help us to change our data distribution.

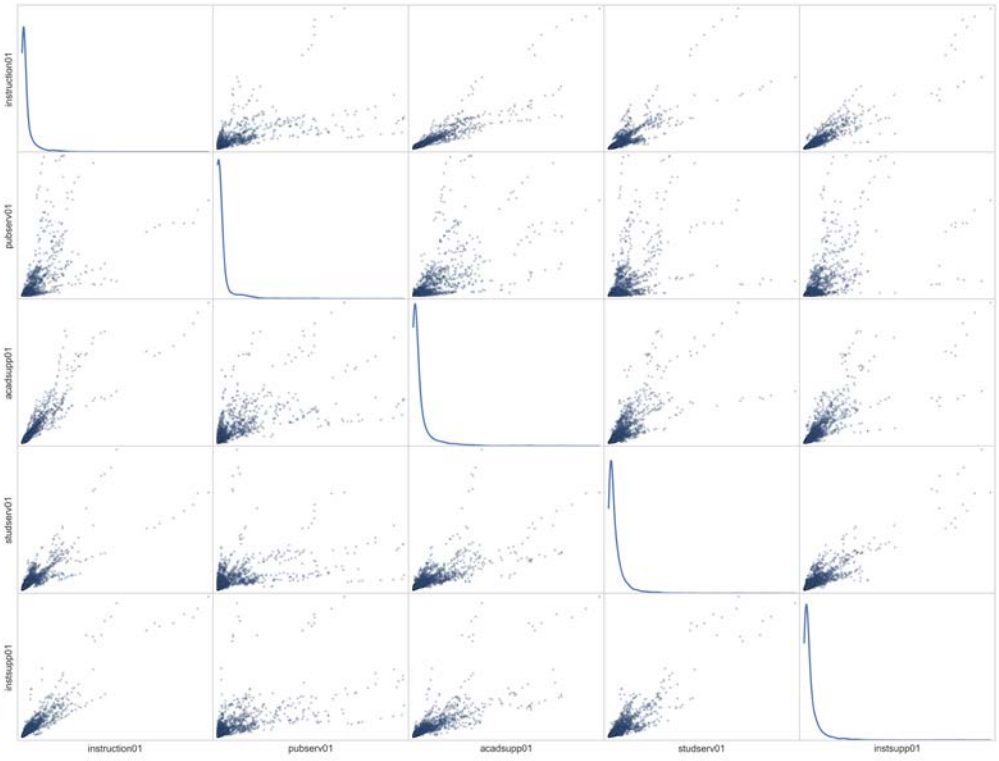

Figure 2- Scatter matrix of first five features. All the features are highly skewed.

Transforming Skewed Features

If data is not normally distributed, especially if the mean and median vary significantly (indicating a large skew), it is most often appropriate to apply a non-linear scaling, particularly for financial data. One way to achieve this scaling is by using a Box-Cox test, which calculates the best power transformation of the data that reduces skewness. A simpler approach which can work in most cases would be applying the natural logarithm where it will be implemented in this section.

Figure 3 shows the log-transformed data, the scatter matrix of first five features after applying a natural logarithm scaling to the data. The distribution of each feature appears much more normal. For any pairs of features that have been identified earlier as being correlated, the correlation is still present and it is now stronger and more clear than before.

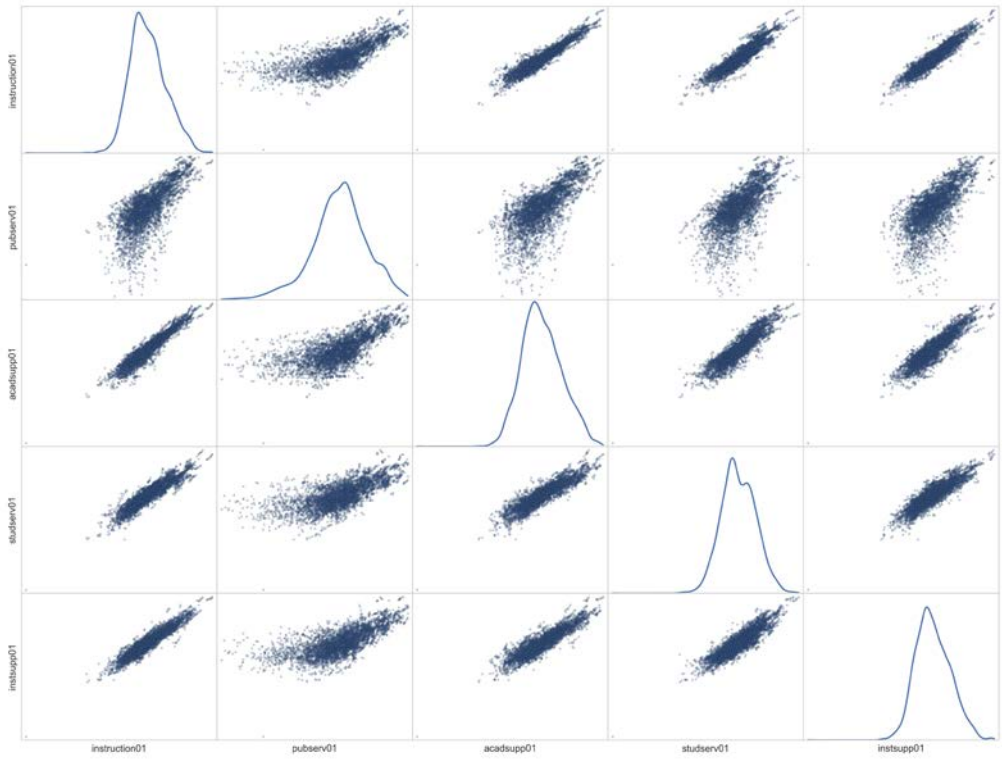

Figure 3- Transformed data after applying the natural logarithm. (first five features)

Algorithms and Techniques

So far, the datasets have been loaded with *python pandas* library and stored as a data frame. After removing outliers, missing values, unnecessary data points and features, the log transformation of data was implemented. Next, cross-validation function will be used to split the data into training and the testing datasets (75% assigned to training datasets and 25% assigned to testing datasets).

Having the training and testing datasets ready, the Principal Component Analysis (PCA) will be implemented to find the dimensions that explain the most of the variance. The top components become predicting features for creating next models. Before the PCA procedure, the standardizing and feature scaling procedures will be performed on the dataset.

Different algorithms will be used to create predictive models. These models will be able to predict a specific institution completion rate based on its expenditure and financial aid features. The following are few supervised learning models that are currently available in *scikit-learn* that from which four algorithms will be selected and studied further in this project.

- **K-Nearest Neighbor Regressor**: In k-NN regression, the k-NN algorithm is used for estimating continuous variables. One such algorithm uses a weighted average of the k nearest neighbors, weighted by the inverse of their distance. A peculiarity of the k-NN algorithm is that it is sensitive to the local structure of the data, and the model costs less with minimal tuning.
- **Decision Tree Regressor**: Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with

decision nodes and leaf nodes. The model is easy to implement and also performs very fast. Once trained the model predicts very quick. The model gives a clear structure of how it makes a prediction, but the downside is the prediction may lack interpretation.

- **Support Vector Regressor**: A version of SVM for regression was proposed in 1996 by Vladimir N. et al. is called support vector regression (SVR). The model produced by support vector classification depends only on a subset of the training data, because the cost function for building the model does not care about training points that lie beyond the margin. Analogously, the model produced by SVR depends only on a subset of the training data, because the cost function for building the model ignores any training data close to the model prediction. SVR is less prone to over-fitting, and need less tuning.
- **Multiple Layer Perceptron Regressor**: The model uses the structure of a neural network and is regaining popularity due to numerous applications on voice and image recognition. MLPs are universal function approximators as showed by Cybenko's theorem, so they can be used to create mathematical models by regression analysis. The term "multilayer perceptron" does not refer to a single perceptron that has multiple layers. Rather, it contains many perceptrons that are organized into layers. Moreover, MLP "perceptrons" are not perceptrons in the strictest possible sense. True perceptrons are formally a special case of artificial neurons that use a threshold activation function such as the Heaviside step function. MLP perceptrons can employ arbitrary activation functions. A true perceptron performs binary classification (either this or that), an MLP neuron is free to either perform classification or regression, depending upon its activation function.
- **Random Forest Regressor**: The model creates multiple decision trees and trains with data through bootstrap sampling. Every node on the branch will randomly choose a small amount of the features. The trees will be tested with data not sampled. This method can avoid over-fitting to training data by its randomness. The setting has ten decision trees as estimators with no maximum depth and considers all the features when looking for the best split. No maximum for the leaf node and the minimum samples leaf is one. Random decision forests correct for decision trees' habit of overfitting to their training set.

In the next step, the models will be optimized and tuned using Grid Search function and sets of hyperparameters. The metrics mentioned before will be imported from *scikit-learn* package and will be used to evaluate different models and their testing results.

The final goal would be finding the most accurate model which is capable of predicting target label (completion rates) with highest performance scores. An important task when performing supervised learning on a dataset like this is determining which features provide the most predictive power. By focusing on the relationship between only a few crucial features and the target label, one could simplify the understanding of the phenomenon, which is almost always a useful thing to do. Moreover, using the Principal Component Analysis, a list of expenditure and financial aid features will be selected to be used as inputs for the final model in order to speed up the prediction process while maintaining the accuracy in an acceptable range.

Benchmark Model

An un-tuned *AdaBoostRegressor* is used as the Benchmark Model. This model is trained and tested on the original training and testing datasets (datasets before implementing the log-transformation). Next, the selected models will be trained using PCA-transformed datasets and will be tuned and

optimized further. The evaluation metrics and performance scores (R^2) will be used to compare the results of the benchmark model against other models and the optimized ones.

Data Preprocessing

In addition to data cleaning procedures and performing log-transformation on features that are highly skewed, it is often good practice to perform some type of feature scaling on numerical features.

Feature Scaling

Applying a scaling to the data does not change the shape of each feature's distribution, however, normalization ensures that each feature is treated equally when applying supervised learners. Moreover, once scaling is applied, observing the data in its raw form will no longer have the same original meaning. *sklearn MinMaxScaler* is used to normalize each numerical feature.

Typically, learning algorithms expect input to be numeric, which requires that non-numeric features (called categorical variables) be converted into a "dummy" variables. Since there are no features that are non-numeric, the encoding step will be skipped.

PCA

In this section, Principal Component Analysis (PCA) is used to draw conclusions about the underlying structure of the Expenditure/Financial datasets. Since using PCA on a dataset calculates the dimensions which best maximize variance, it shows which compound combinations of features best describe the target label.

Now that the data has been scaled to a more normal distribution and has had any necessary outliers removed, PCA is applied to the dataset to discover which dimensions about the data best maximize the variance of features involved. In addition to finding these dimensions, PCA will also report the explained variance ratio of each dimension, i.e. how much variance within the data is explained by that dimension alone. Note that a component (dimension) from PCA can be considered a new "feature" of the space however, it is a composition of the original features present in the data. **Figure 4** shows the normalized explained variance ratio (weights) both incrementally and cumulatively.

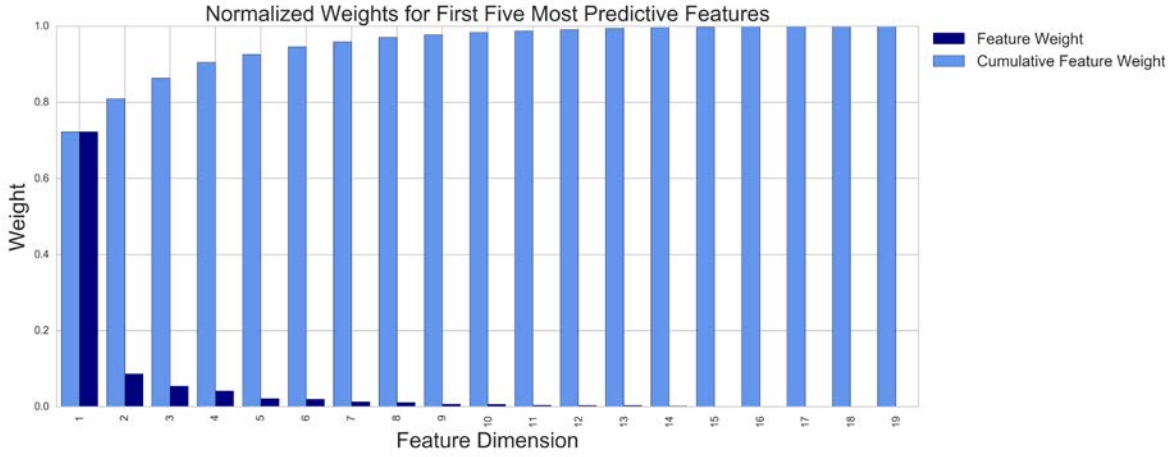

Figure 4- Normalized Explained Variance Ratio (Weights) for all PCA dimensions

Dimensionality Reduction

Based on **Figure 4**, 81% of the variance in the data is explained in total by the first and second principal component and 10% by third and fourth principal components. Therefore, we can say 91% of the variance in the data is explained by the first four dimensions.

When using principal component analysis, one of the main goals is to reduce the dimensionality of the data, in effect, reducing the complexity of the problem. Dimensionality reduction comes at a cost: Fewer dimensions used implies less of the total variance in the data is being explained. Because of this, the cumulative explained variance ratio is extremely important for knowing how many dimensions are necessary for the problem. Additionally, if a significant amount of variance is explained by only two or three dimensions, the reduced data can be visualized afterward.

In this study, we chose to reduce the dimensionality to seven principal components since 97% of the variance in the data is explained in total by the first seven components. **Figure 5** shows the results of assigning and fitting PCA in seven dimensions with the final dataset.

Note that a positive increase in a specific dimension corresponds with an increase of the positive-weighted features and a decrease of the negative-weighted features. The rate of increase or decrease is based on the individual feature weights.

Using the visualization below, one can see the first dimension best represent all the expenditure and financial aid categories. The second dimension is totally in a different direction and mostly depends on **loan_num** feature.

The third dimension differentiates between the effect of **pubserv01** and **rschpub01** versus the rest of features and the fourth dimension further accentuate the difference between **other01** versus **pubserv01** and **rschpub01**.

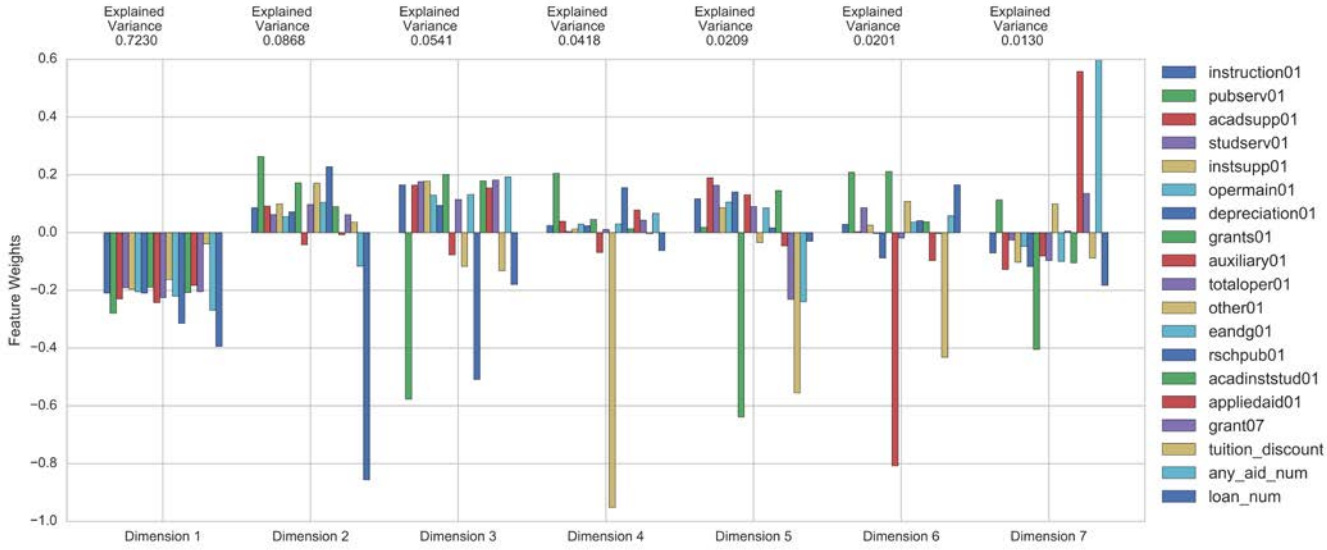

Figure 5- Results of fitting PCA in seven dimensions with the final dataset.

Implementation

From the mentioned list of the supervised learning models, four algorithms that are more appropriate for regression analysis of this project have been selected and will be tested on the IPDES dataset. These algorithms are; *DecisionTreeRegressor*, *KNeighborsRegressor*, *AdaBoostRegressor*, and *RandomForestRegressor*. These algorithms can handle lots of irrelevant features and perform well with large datasets. Other models aren't suitable for high-dimensional cases and are prone to overfitting.

Creating a Training and Predicting Pipeline

To properly evaluate the performance of each selected model, it's important to create a training and predicting pipeline that allows us to quickly and effectively train models using various sizes of training data and perform predictions on the testing data. The implementation here will be used in the following section.

In the predicting pipeline, at first, the score metrics are imported from *sklearn*. Then the learner will be *fit* to the sampled training data and the training time will be recorded. Next, predictions on the testing dataset and also on the first 300 training data points will be performed and the total prediction time will be recorded. Finally, R^2 score for both the training subset and testing set will be calculated.

Initial Model Evaluation

Four supervised learning models discussed in the previous section are imported from *sklearn* and initialized and stored in *Reg_A, Reg_B, Reg_C,* and *Reg_D*. A *random_state* and the default settings for each model is used. (tuning and optimization will be done on specific models in next sections). Samples of training data (equal to 1%, 10%, and 100% of the training data) are used for training each model. Finally, the models are tested using the testing dataset and the results are collected in **Figure 6**.

The coding for the predicting pipeline is straightforward however these points are noteworthy;

- The prediction time was chosen in a way to include both training and testing datasets and capture their effects. This way the evaluation metric is more accurate and generalized.
- To have a more comprehensive understanding of the effect of training data size on the final results, samples of training data (equal to 1%, 10%, and 100% of the training data) are used for training each model. One can examine different subsets of data to find the point where beyond that, the R^2 are not changing significantly.
- In this study, we have used the original label values (y values) for training and testing procedures. If we had multiple features as a label, we should have used the transformed label values (y-log-transformed) in the training and testing steps.

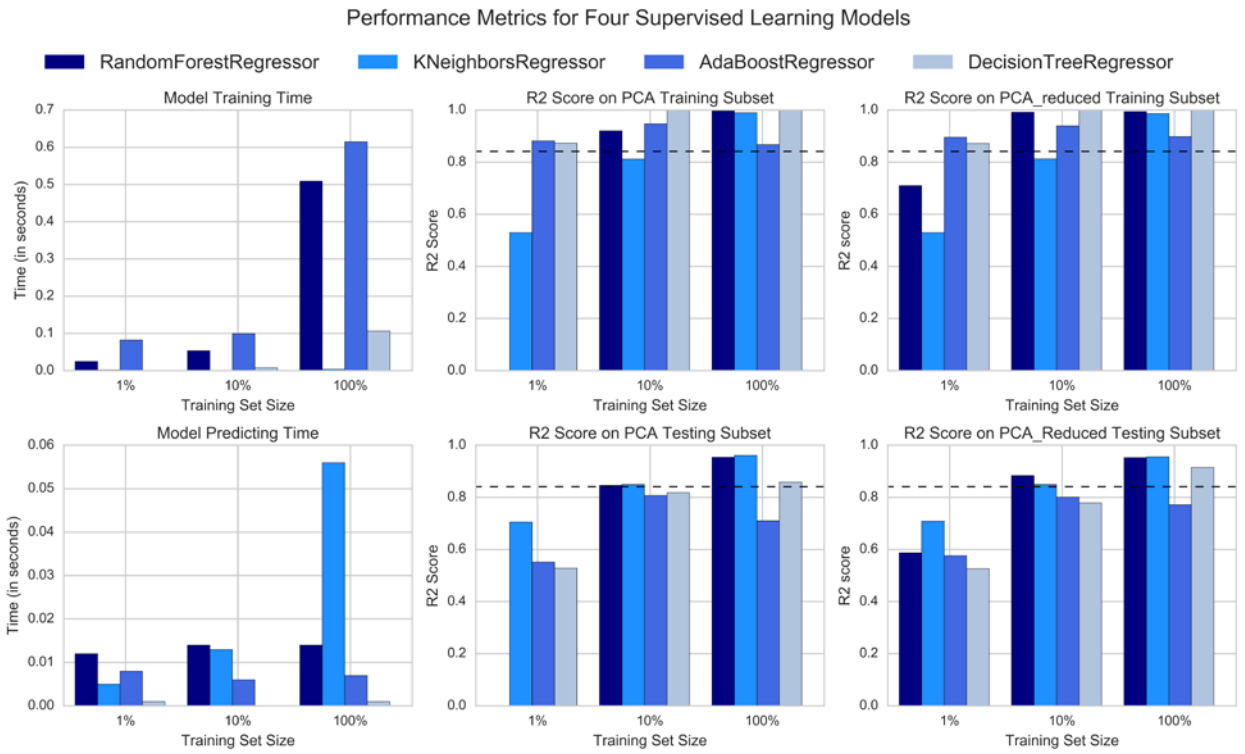

Figure 6- Performance metrics for four supervised learning models.

Choosing the Best Model

Based on our evaluation, *RandomForestRegressor* model seems to be the most appropriate one for the task of predicting institutes completion rates using Expenditure/Financial data features. *KNeighborsRegressor* model has the highest R^2 score on both PCA and reduced PCA testing sets when 100% of the training set is used (0.96 and 0.95 respectively). *RandomForestRegressor* has almost the same R^2 score results (0.95 and 0.94 respectively) but it has a significantly shorter testing time. On the other hand, *KNeighborsRegressor* has shorter training time which makes it very competitive with *RandomForestRegressor* and the second choice in terms of efficiency. Moreover, *DecisionTreeRegressor* has lowest performance results although it has the fastest training and testing time.

Refinement

Model Tuning using Grid Search and Cross-Validation

In this section, the chosen models (*RandomForestRegressor* and *KNeighborsRegressor*) are optimized and fine-tuned. *k-fold* cross validation split the training set into *k* bins, use a bean as testing data and use rest of the data as training data and validate against testing data. It repeats the process k times. The performance measure reported by *k-fold* cross validation is then the average of the values computed in the loop. *k-fold* cross validation especially useful for small dataset since it maximizes both test and training data.

This technique is very useful for grid search when optimizing a model. The grid search, systematically working through multiple combinations of hyper-parameters, tunes and determine which one gives the best performance.

In this project, grid search function (*GridSearchCV*) is used with a couple of important parameters (hyper-parameters) tuned with at least 3 different values. A dictionary of hyper-parameters is created to be used in grid search procedure for the chosen models and the entire training set are being used for this process. **Table 3** shows all the hyperparameters and their respective values tried within the tuning process.

Table 3- Hyperparameters and their respective values assigned for tuning of both models.

RandomForestRegressor		
n_estimators	The number of trees in the forest.	[10,50,100,200,500]
max_features	The number of features to consider when looking for the best split.	['auto', 'sqrt', 'log2']
max_depth	The maximum depth of the tree.	[10, 20, 30, 50]
min_samples_split	The minimum number of samples required to split an internal node.	[2, 4, 8]
bootstrap	Whether bootstrap samples are used when building trees.	[True, False]
KNeighborsRegressor		
n_neighbors	Number of neighbors to use by default for kneighbors queries.	[1,3,5,10,20,50,70,100]
weights	weight function used in prediction.	['uniform','distance']
algorithm	Algorithm used to compute the nearest neighbors.	['auto', 'ball_tree', 'kd_tree', 'brute']
leaf_size	Leaf size passed to BallTree or KDTree. This can affect the speed of the construction and query, as well as the memory required to store the tree.	[1,5,10,30,50]

Justification

Table 4 shows the optimized model's R^2 score on the PCA reduced testing data along with the results from our unoptimized model and the benchmarks model discussed earlier.

Table 4- R2 score of optimized and unoptimized models along with benchmark R2 score.

	Benchmark Predictor	Unoptimized Model	Optimized Model
RandomForestRegressor	0.8413	0.9494	0.9552
KNeighborsRegressor	0.8413	0.9561	0.9636

The above table shows optimized model's score on the testing data are 0.9552 and 0.9636 respectively which are slightly better than the unoptimized model. The hyperparameters tuning using grid search (GridSearchCV) were effective for our models. Our optimized models performed significantly better than the benchmark model. The R^2 score changed from 0.8413 to 0.9552 and 0.9636.

Table 5 includes the detail of optimized and tuned hyperparameters for the first choice model, *RandomForestRegressor* and second choice model, *KNeighborsRegressor*.

Table 5- Optimized and tuned hyper parameters of first and second choice models.

KNeighborsRegressor	*RandomForestRegressor*
KNeighborsRegressor(algorithm='auto', leaf_size=1, metric='minkowski', metric_params=None, n_jobs=1, n_neighbors=3, p=2, weights='distance')	RandomForestRegressor(bootstrap=False, criterion='mse', max_depth=50, max_features='log2', max_leaf_nodes=None, min_impurity_split=1e-07, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=1, oob_score=False, random_state=None, verbose=0, warm_start=False)

Model Evaluation and Validation

To validate the robustness of the model, two well-formed tests are performed on the final model to evaluate the sensitivity of model to;

- Outliers in the data
- Small changes or perturbations in the data

For the first step, the final tuned *RandomForestRegressor* model is trained and tested on the original training and testing dataset before the transformation and outlier removal. The R^2 score is 0.9575 which is 0.63% lower than the model trained on PCA_reduced training dataset and comparing with the benchmark, R^2 score is 13.81% higher. This shows that the selected model is not sensitive to the outliers.

For the second step, the random state variable of the algorithms (used to sequester the data and construct the model) is altered. To assure the test results are reliable it is critical to prevent information from the testing data being used in the construction of the model. By keeping the random state constant, the training and testing subsets are always the same, and no information leak occurs. Once the model is finalized, the sensitivity of the model to the input space can be tested by altering the random state and thus the subsets of data with which the model is built and tested. This results in a model with a very close R^2 score of 0.9632, showing the model is resilient to perturbations in the data.

As a result, the final tuned model generalizes well to unseen data and is not sensitive to small changes in the data, or to outliers and can be trusted.

Conclusion

Free-Form Visualization

The first objective of this project was to find out whether the completion rates could be predicted based on financial aid and expenditure data from post-secondary education institutions and which predictive model can handle this task with the highest performance score.

Implementing PCA procedure, the 19 features that we selected and processed from original IPEDS dataset, were reduced to 7 dimensions. Using this transformed data, different models were tested on the testing dataset and *RandomForestRegressor* showed the best performance among all candidates in terms of testing time and R^2 score. As a result, this model is selected as the best model in terms of predicting target labels (institutes completion rates).

As the second objective of this project, an important task when performing supervised learning on a dataset like the education dataset under study here is determining which features provide the most predictive power. By focusing on the relationship between only a few crucial features and the target label we simplify our understanding of the phenomenon, which is almost always a useful thing to do. In the case of this project, that means we wish to identify a few number of features that most strongly predict institutes completion rates.

Using PCA analysis performed before and based on **Figure 5**, 97% of the variance in the data is explained in total by the first seven principal component. The significant positive increase in a specific dimension corresponds with an increase of the positive-weighted features and a decrease of the negative-weighted features. The rate of increase or decrease is based on the individual feature weights. Using the visualization, one can see the first dimension best represent all the expenditure and financial aid categories. The second dimension is totally in a different direction and mostly depends on **loan_num** feature. The third dimension differentiates between the effect of **pubser01** and **rschpub01** versus the rest of features and the fourth dimension further accentuate the difference between **other01** versus **pubser01** and **rschpub01**. These features are most strongly predicting the institutes completion rates. **pubser01** is the expense category that provides noninstructional services beneficial to individuals and groups external to the institution such as conferences. **loan_num** is the number of full-time, first-time degree/certificate-seeking undergraduate students who received student loans. Other features can be looked up in **Table 1**.

To better understand the feature importance and being able to visualize the feature importance metrics, we create a PCA with just two components representing two dimensions. Remember that the first two component explains 81% of the variance in the data. In addition, as we discussed above, only top six features are used to train and fit the new created PCA. **Figure 7** shows the biplot which is a scatterplot where each data point is represented by its scores along the principal components. The axes are the principal components (in this case Dimension 1 and Dimension 2). In addition, the biplot shows the projection of the original features along the components. A biplot can help us interpret the reduced dimensions of the data, and discover relationships between the principal components and original features.

Once we have the original feature projections (in blue), it is easier to interpret the relative position of each data point in the scatterplot. From the biplot, we can see the original features of **pubser01, rschpub01, other01, grants01,** and **tuition_discount** are most strongly correlated with the first component and **loan_num** is associated with the second component. Moreover, these observations agree with the pca_results plot we obtained earlier. The length of each arrow shows the importance

level of each feature. **loan_num, pubser01** and **rschpub01** are the first, second and third important features in our study respectively.

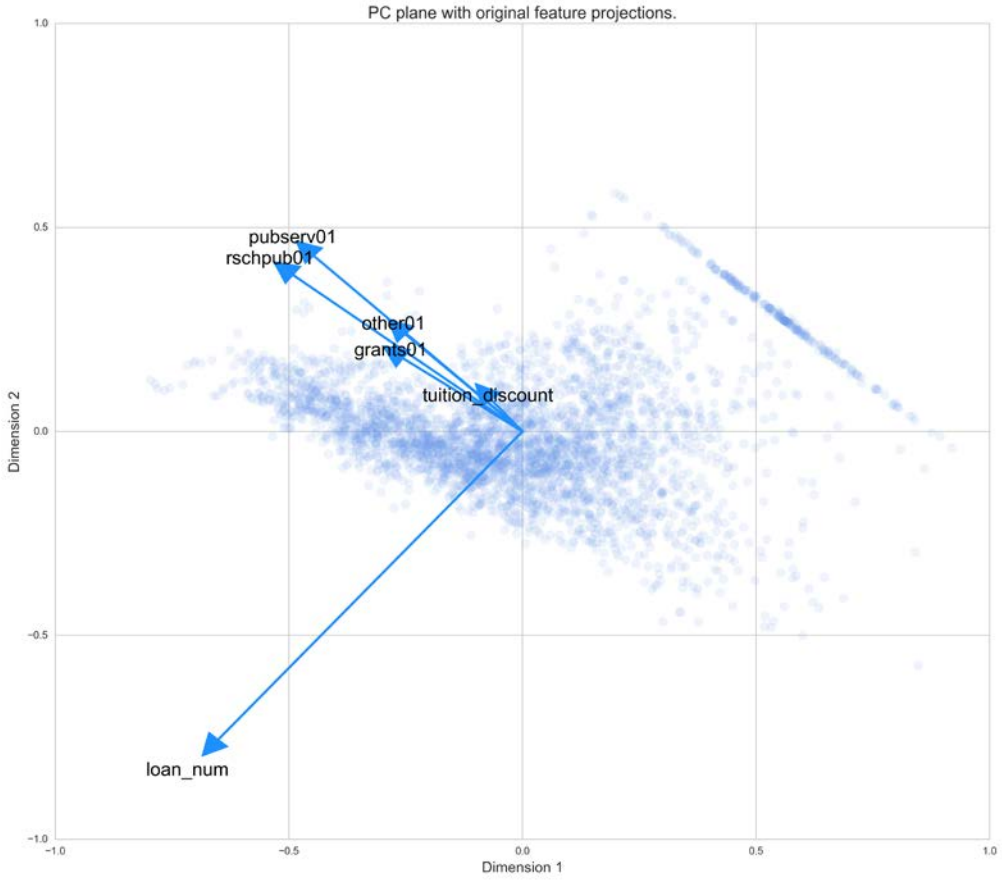

Figure 7 - biplot based on 2 component PCA analysis and six top features.

Reflection

This project focused on studying and analyzing different components of the expenditure and financial aid of higher education institutes and utilized them to predict the total number of awards, certificates, and degrees completed each year in these institutions. The project successfully used different tools such as PCA to find the effective combination of features and tested and optimized different algorithms to come up with a model capable of performing better than the benchmark model and other candidates.

Data cleaning and selecting proper features for analysis were the most time-consuming part of this project. For example, some institution runs private labs and hospitals while others have collaboration with industrial or tech companies and finding a common ground to make feature selection was complicated. Many fields excluded for being not relevant to this study and lots of data points with missing values were deleted eventually. Finding suitable algorithms for regression analysis was also struggling. While some well-known ensemble methods (such as Random Forest) seems to be a good fit but their characteristics and specific behavior of these models might not be the best option when dealing with IPEDS dataset.

Improvement

The project has lots of room for improvement. Rather than predicting the rankings of institutes using financial and expenditure data, one can include or exclusively use other features such as revenue, number of faculties and employees, geographic region, census division, Carnegie classification, and years to graduation. Moreover, other categories of expenditure and financial aid that has not been included in this study can be considered.

In addition, different data cleaning approaches or data transformation methods could be examined further. PCA reduces problem dimensionality into a specific number of principal components. With further investigation, one might be able to run multiple scenarios with different combinations of components and see its effect on model performances. Using other algorithms such as *Gradient Boosting Regressor* or *Support Vector Regressor* for regression analysis along with more effective grid search procedure could be studied further as well.

References

1. Integrated Postsecondary Education Data System Delta Cost Project Database, https://nces.ed.gov/ipeds/deltacostproject/
2. Rankings of universities in the United States, https://en.wikipedia.org/wiki/Rankings_of_universities_in_the_United_States
3. 2009–2010 College Rankings: National Universities, http://www.parchment.com/
4. Statistics and Probability Dictionary, http://stattrek.com/statistics/dictionary.aspx?definition=coefficient_of_determination
5. "Applied to Stanford or Harvard? You probably didn't get in. Admit rates drop, again.". Retrieved 13 May 2016. https://www.washingtonpost.com/news/grade-point/wp/2016/04/01/
6. Drucker, H. (1997, July). Improving regressors using boosting techniques. In ICML (Vol. 97, pp. 107-115).
7. Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of machine learning, research, 3(Mar), 1157-1182.
8. National Conference of State Legislatures (2015, July 31). Performance-based Funding for Higher Education Retrieved from http://www.ncsl.org/research/education/performance-funding.aspx