Taylor Tucker, Tina Jin, Virginia Weston, Jeffrey Bradley
Prof. Watson
CSCI 297
21 November 2020

Predicting the Total Cost of Attendance of Universities Using Machine Learning Techniques

Abstract
Cost of attendance has become one of the most important factors when students choose universities to attend. We propose a method of predicting the total cost of attendance of U.S. universities using seven machine learning models. Our method could advise schools on setting appropriate amount of tuition, and could help students predict the total cost of a university based on its key statistics.
Keywords: tuition, data analysis, machine learning, regressors, classifiers

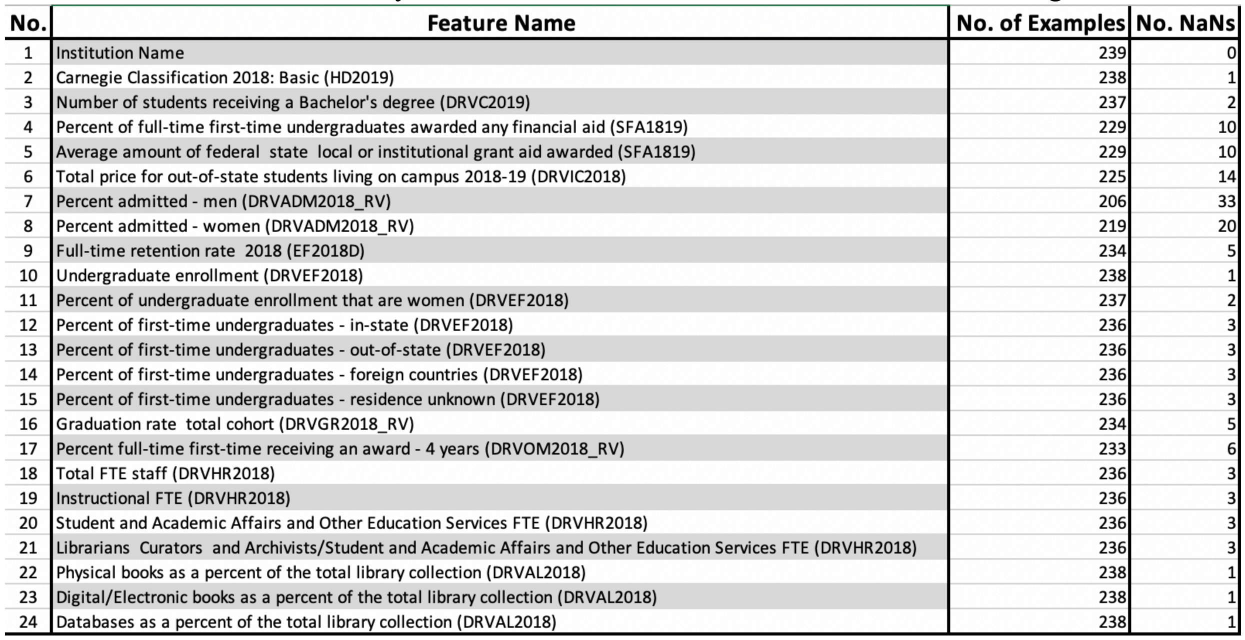
1. Introduction

1.1 Problem

The cost of attendance of U.S. universities is an ever-growing issue for many teenagers and their families. Over the last decade, the average cost of attendance for U.S. colleges has increased more than 25% (Hess). Not only do prospective students wonder about their or their families' ability to pay, but also where all that money goes. We propose a method of using both machine learning regressors and classifiers to both classify a college's total cost of attendance into price brackets, or predict the tuition.

1.2 Data Gathering

We gathered our data from the open source Integrated Postsecondary Education Data System, which is a part of the National Center for Education Statistics. We manually selected the types of colleges we wanted, were relatively small baccalaureate colleges and universities focused on arts and sciences. We then manually selected the features we wanted to use, seen in Figure 1.



No.	Feature Name	No. of Examples	No. NaNs
1	Institution Name	239	0
2	Carnegie Classification 2018: Basic (HD2019)	238	1
3	Number of students receiving a Bachelor's degree (DRVC2019)	237	2
4	Percent of full-time first-time undergraduates awarded any financial aid (SFA1819)	229	10
5	Average amount of federal state local or institutional grant aid awarded (SFA1819)	229	10
6	Total price for out-of-state students living on campus 2018-19 (DRVIC2018)	225	14
7	Percent admitted - men (DRVADM2018_RV)	206	33
8	Percent admitted - women (DRVADM2018_RV)	219	20
9	Full-time retention rate 2018 (EF2018D)	234	5
10	Undergraduate enrollment (DRVEF2018)	238	1
11	Percent of undergraduate enrollment that are women (DRVEF2018)	237	2
12	Percent of first-time undergraduates - in-state (DRVEF2018)	236	3
13	Percent of first-time undergraduates - out-of-state (DRVEF2018)	236	3
14	Percent of first-time undergraduates - foreign countries (DRVEF2018)	236	3
15	Percent of first-time undergraduates - residence unknown (DRVEF2018)	236	3
16	Graduation rate total cohort (DRVGR2018_RV)	234	5
17	Percent full-time first-time receiving an award - 4 years (DRVOM2018_RV)	233	6
18	Total FTE staff (DRVHR2018)	236	3
19	Instructional FTE (DRVHR2018)	236	3
20	Student and Academic Affairs and Other Education Services FTE (DRVHR2018)	236	3
21	Librarians Curators and Archivists/Student and Academic Affairs and Other Education Services FTE (DRVHR2018)	236	3
22	Physical books as a percent of the total library collection (DRVAL2018)	238	1
23	Digital/Electronic books as a percent of the total library collection (DRVAL2018)	238	1
24	Databases as a percent of the total library collection (DRVAL2018)	238	1

Figure 1: Features Gathered from IPEDS

Taylor Tucker, Tina Jin, Virginia Weston, Jeffrey Bradley
Prof. Watson
CSCI 297
21 November 2020

1.3 Data Cleaning

To clean the data we gathered, we first had to decide what to do about the missing data. As seen in Figure 1, most of the missing data belongs to the features "Percent admitted – men" and "Percent admitted – women", likely because of the prevalence of single-gender universities or because of a lack of reporting. To fix this, we revisited the IEPDS website and gathered the admissions data for both sexes combined. This left us with only a few NaNs, which we imputed using the features median value. We continued to drop each example that had a NaN value in the target vector of "Total Price", as we cannot predict nor accurately impute a target feature without adding more unnecessary correlations to the data. We created simpler header names and dropped the final feature because it was a vector of zeroes; this would not have given us any information gain. This cleaning left us with a dataset containing 19 features, 1 target, and 225 examples.

1.4 Related Works

Otium Advisory Group published an article in their magazine in which they described a statistical approach to calculating the cost of college in the future with a given rate of inflation (Otium Advisory Group). This is simply using statistics for prediction, not machine learning and is related only by comparing college costs. Eric Cheng applied machine learning methods for classifying public versus private institutions in *Classifying colleges with machine learning.* They applied similar models we explored including Decision Trees and KNN. Their dataset was much larger (777 colleges) and they used the decision tree primarily for feature selection and then KNN for classification. They did not discuss accuracy but had low error 0.096, though they did not say what kind of error this was.

2. Method

2.1 Exploratory Data Analysis (EDA)

For our exploratory data analysis, we first began with a scatterplot matrix (**Appendix A**) that showed the relationships between every feature. With 41 features, this plot became very large, however we were able to get an initial idea of how certain features were related and see how some features were very linearly correlated with our target, "Total Cost", such as "Average Amount of Aid", "Retention Rate", "Graduation Rate", and "Percent Awarded [Aid]". Then using a correlation heatmap **(Appendix B),** we found all those mentioned features had a linear correlation of +0.80 with our target. This information could prove useful for regressors, such as linear regression, in determining a simple model to predict a University's cost. Indeed, there were a lot of features that had some sort of relationship with each other. We believe that, due to this data analysis, we have the opportunity to perhaps even create a simple linear model which yields a solid prediction.

Another consideration is how we could do classification using thresholds to divide our data into either binary, higher or lower than a median value, or multiclass such as \$10000 - \$20000, \$20000 – \$30000, etc. Since we do see a lot of relatively correlated features in the heatmap, such as "Graduation Rate" and "Retention Rate", it might be unwise to use a model such as naïve bayes to predict, since many of the features are correlated. However, after feature selection, we might want to evaluate more models than we think. Regardless the ease of model APIs will allow us to apply our data to a variety of models and determine our best results, but our EDA hints that a regression will work best.

2.2 Feature Scaling

For feature scaling, we decided to use Sci-Kit Learn scaling techniques. As seen in the EDA, the effect of scalers on outliers can already be captured by standard and min-max scalers, so we apply these two scalers to the data first. Please see **Appendix D** for examples of how these methods effected our data.

2.3 Dimensionality Reduction

We decided to try out RF feature elimination. For feature selection using random forest, we first applied the standard and min-max scalers to normalize and standardize the dataset. Then we used a random tree regressor on the data with 80-20 train-test split, and printed out the feature importances obtained by the RF. We ranked the 20 features by feature importance, and applied the threshold of 0.01 to the features to get the 8 most relevant ones (**Appendix C**). We also tried out PCA to see if there is any validation in terms of which features we select as the most important.

Following the RF method for dimensionality reduction, we believe it would be beneficial to even reduce the feature space to include only ["Average Amount of Aid", "Percent Financial Aid", "Percent Awarded", "Total Staff", "Graduation Rate", "Percent Admitted", "Number of Bachelor's Degrees"]. We believed this would give us a smaller dimensionality while also maintaining high levels of accuracy.

2.4 Linear Regression

Our linear regression models are used to predict the continuous target variable, Total price. We implemented linear regression with stochastic gradient descent, Lasso, Ridge, and ElasticNet. After splitting the data with 70-30 train-test split, we applied a pipeline of StandardScaler, MinMaxScaler, and then the 4 models. Note that SGDRegressor, Lasso, and Ridge do not contain significant hyperparameters that require a grid search. For the SGDRegressor, we used OLS as the loss function and L2 regularization, which in effect gave us a linear regression model with SGD and L2 regularization. We applied a grid search on ElasticNet to find the optimal hyperparameter 'elasticnetcv__l1_ratio'. After fitting the pipeline to the training data, we ran 10-fold cross validation on the testing data and printed out the accuracies over 10-folds.

2.5 Support Vector Machine

Our Support Vector Machine (SVM) model was used to predict classifications of our target variable "Total Cost" thus correctly classifying a row to be in a cost range such as $50000 - $60000, or 60000 – 70000 etc. We split our data into 70% training and 30% testing, and applied a Standard Scaler to normalize our data. We then ran the entire dataset including all features through a grid search using the SVC sklearn API, C (the regularization parameter) and gamma (the kernel coefficient) ranging from 0.1 to 50.0, and 4 different kernels including linear, rbf, sigmoid, and poly. Our best results came were only 58% training accuracy and 63% testing accuracy with a rbf kernel, C value of 1.0, and gamma value of 0.1. We then tried the same process using our most important features (Average amount of Aid, Graduation Rate) and still only received an 59% training accuracy, and only 44% testing accuracy.

2.6 Decision Tree and Random Forest
We used the Sci-Kit Learn implementation for both the Decision Tree Classifier and the Random Forest Classifier. We generated and used the discreet dataset, with the classifications 10,000-20,000, 20,000-30,000, etc, and split the data on a 70%-30% Train-Test ratio. We then created pipelines using the StandardScaler and MinMaxScaler, and ran a grid search with 10-fold cross validation on the hyperparameters of both the DT and RF. The exact same process was run on the discreet dataset, however, using only the best features that RF feature elimination found. Both classifiers achieved better accuracy when using the entire dataset than when using the few select features from RFFE. The best training accuracies for DT and RF were 52.8% and 61.1% respectively, and the best testing accuracies for DT and RF were 61.8% and 63.2% respectively. The best hyperparameters found by the cross-validated grid search for DT and RF were [criterion = "entropy", max_depth = 5] and [criterion = "gini", max_depth = 200, n_ests = 200], respectively.

2.8 K-Nearest Neighbor
To produce our KNN model, we found it optimal to split the data into 70% training and 30% testing. With this split data, we tested and compared several scalers such as PowerTransformer, StandardScaler, MaxAbsScaler, and RobustScaler, finding that the PowerTransformer produced the best results because it maintained spacial relationships while making the features more Gaussian distributed. We ran scikit's KNeighborsClassifier on different n_neighbors one through 10 and tested different distance metrics Euclidean and Manhattan. Implementing a KNN with 10 n_neighbors and a Manhattan distance metric produced the highest testing accuracy for this dataset of 60.3%.

2.9 Naïve Bayes
Due to the relatively high correlation amongst numerous features and our target variable, Total Price, we previously inferred that a probability model like Naïve Bayes would not perform well. As expected, the implementation of a Gaussian Naïve Bayes model had weak performance with only 64.4% accuracy. We compared the accuracies produced by Gaussian, Bernoulli, Multinomial, and Complement models on the seven best features, and we found that a Gaussian implementation outperformed the other three model types by over 20 percent. Running a Gaussian model with all of the features resulted in a lower 62.2% accuracy.

3. Results

3.1 Training accuracies and R^2 scores for classifiers and regressors respectively

Model	**Linear Regression**	**SVM**	**Naïve Bayes**	**KNN**	**Decision Tree**	**Random Forest**
Best Train Accuracy/ R^2	**80%**	**58%**	**65%**	**54.2%**	**52.8%**	**61.1%**
Best Test Accuracy/ R^2	**84.6%**	**63%**	**64.4%**	**60.3%**	**61.8%**	**63.2%**

3.2 Conclusion

We decided that the linear regression models using the best two features (i.e., "Average Amount of Aid" and "Graduation Rate") and L2 regularization is the best regressors we could get. As seen in the table above, the testing accuracy of 84.6%, even with L2 regularization and feature scaling.

This accuracy is much better than the accuracy on any of our classifiers. We think that using only those two features with L2 regularization would help us get the best model for the masses because it would decrease the generalization error that could come from using all of the features. Also, if using 2 features can achieve nearly the same effect as using 20, we would stick to them to save computation time.

4. Discussion

Based on our EDA and feature extraction, we originally believed a regression model would perform best and Naïve Bayes would perform worst based on the high correlation between features. We also predicted, due to the few features that had strong linear correlation to our target, that linear regression could be a simple yet effective model. As the results reveal, our hypothesis was on track but we found that all classifiers performed similarly while our linear regression performed well above them. Note that the classifiers try to solve a multiclass classification problem with 7 classes, so a ~60% accuracy is not terrible either. Moreover, we would have liked to use other metrics than accuracy, however, Sci-Kit Learn does not allow for using metrics such as recall or F1 scores with non-binary classification models.
The linear regression model is also well-suited to our purpose of predicting the price as a continuous variable, instead of a multiclass discrete variable. Also note that our testing accuracies are almost always better than the training accuracies, so we would expect that our final model generalizes well.

This generalizability arises from our investigation, wherever possible, into the optimal hyperparameters of the model using a cross-validated grid search, feature scaling, regularization using L1 and L2 regularization, and dimensionality reduction from our EDA.

Indeed, the success of the Linear Regression model based on only two of the features lends itself to a discussion of Occam's Razor: there are variables which correlate strongly with the total price of college attendance. Therefore, there is likely an opportunity available to predict the cost of a university using strictly statistical analysis of these variables without the need for these relatively complex learning algorithms.

Taylor Tucker, Tina Jin, Virginia Weston, Jeffrey Bradley
Prof. Watson
CSCI 297
21 November 2020

Works Cited

Hess, Abigail. “The Cost of College Increased by More than 25% in the Last 10 Years Here’s Why.” *CNBC*, 13 Dec. 2019.

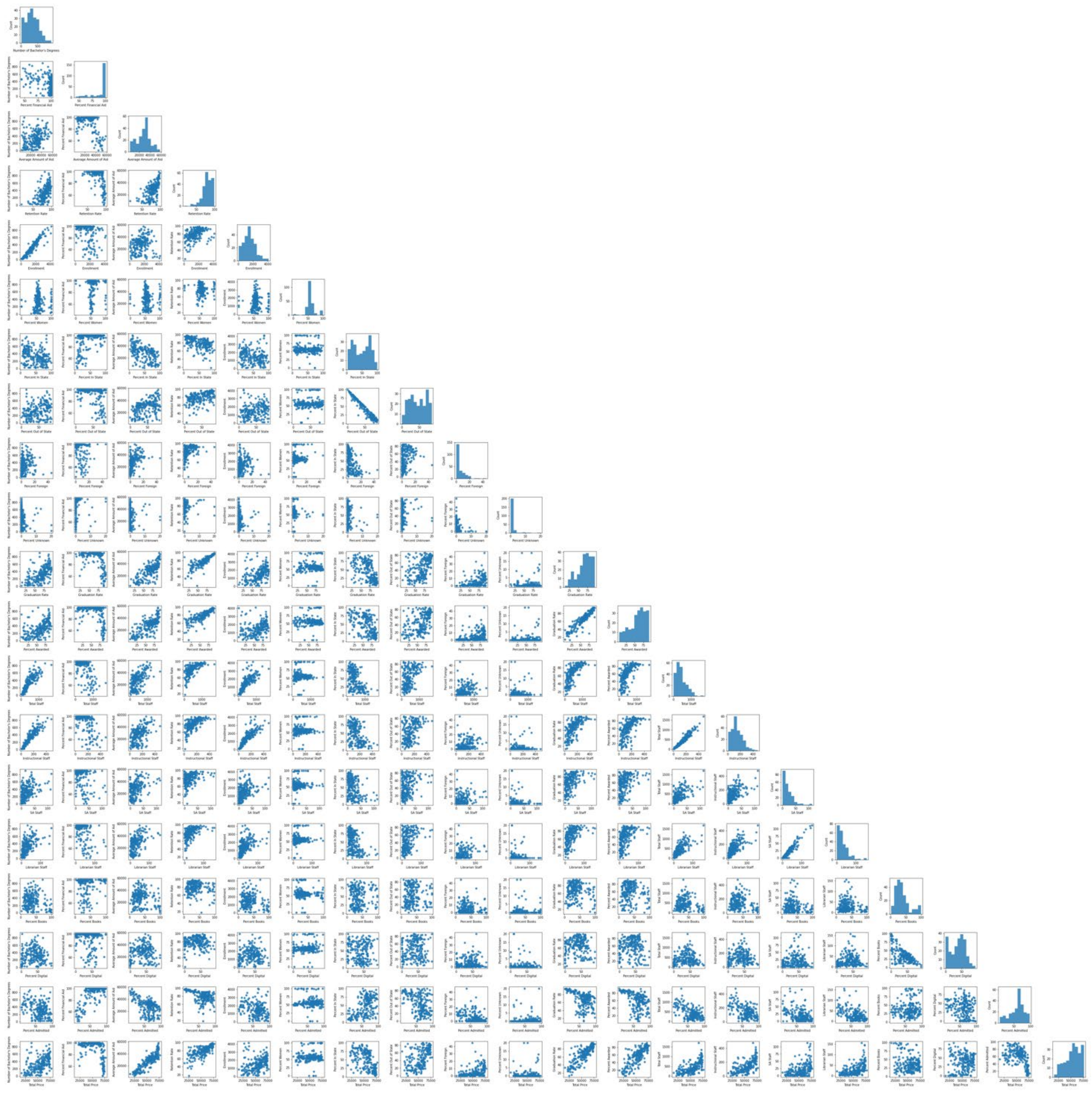
Integrated Postsecondary Education Data System. National Center for Education Statistics.

Otium Advisory Group. “Estimating the Future Cost of a College Education.” *OTIUMAG*.
Raschka, Sebastian, and Vahid Mirjalili. *Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn, and TensorFlow 2*. 2019. *Open WorldCat*, http://proquest.safaribooksonline.com/?fpi=9781789955750.

Cheng, Eric. “Classifying colleges with machine learning” *Carnegie Melon University*. 18 Jul 2018, https://chengeric.com/posts/classifying-colleges-with-machine-learning.

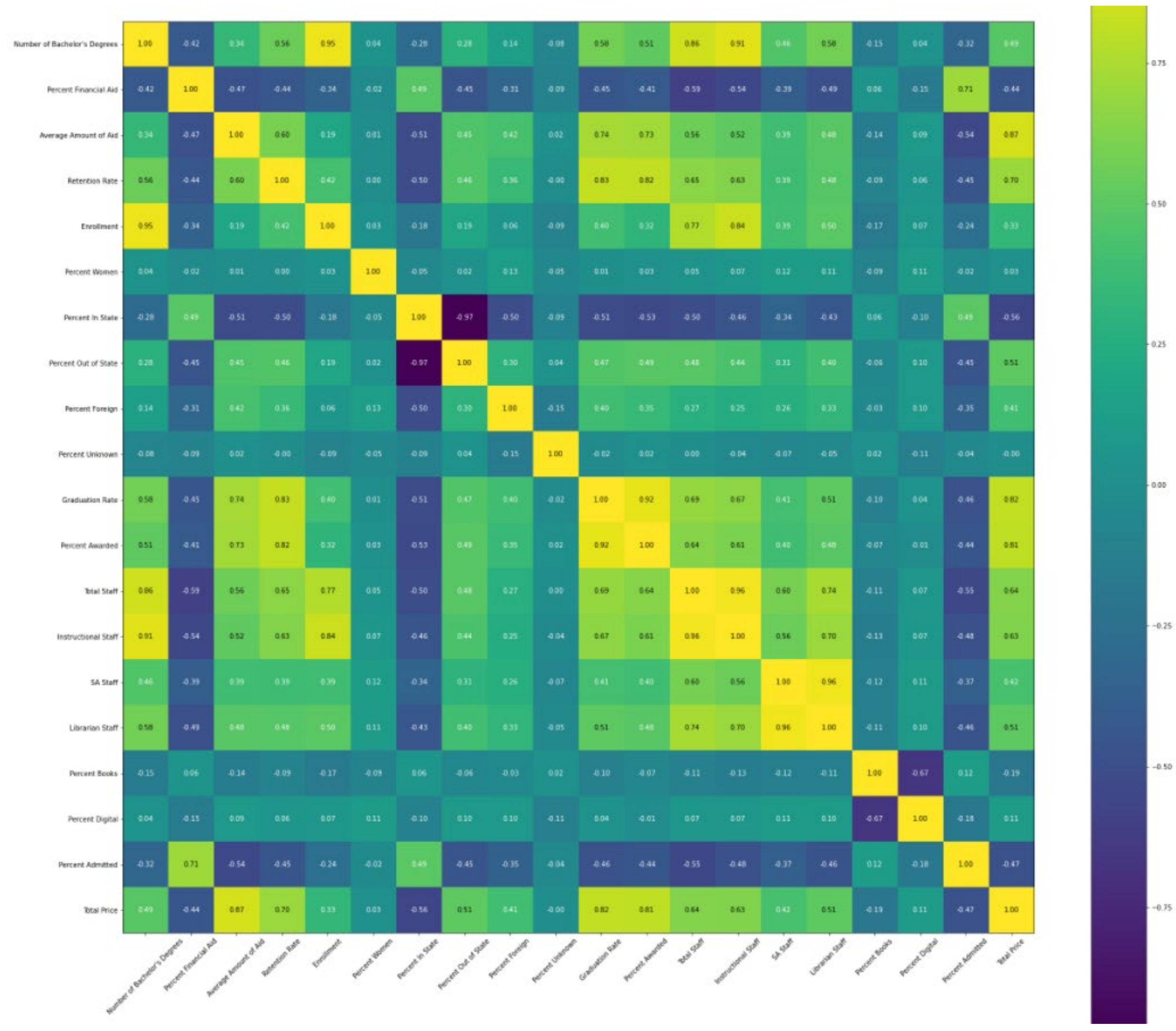
Taylor Tucker, Tina Jin, Virginia Weston, Jeffrey Bradley
Prof. Watson
CSCI 297
21 November 2020

Appendix A
Scatter Plot Matrix from EDA



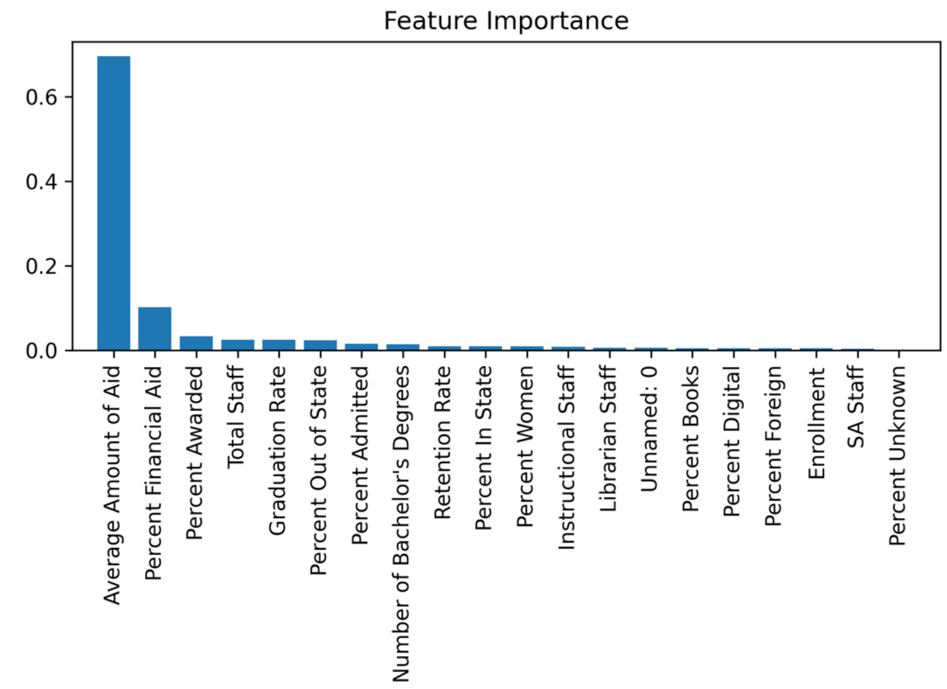
Taylor Tucker, Tina Jin, Virginia Weston, Jeffrey Bradley
Prof. Watson
CSCI 297
21 November 2020

Appendix B
Heatmap from EDA

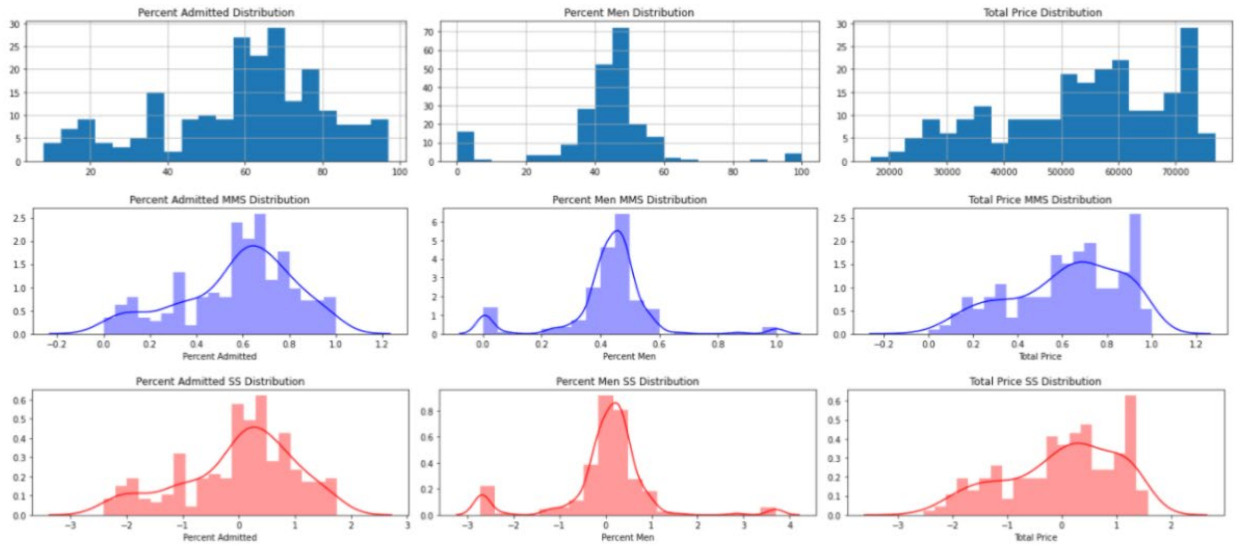


Taylor Tucker, Tina Jin, Virginia Weston, Jeffrey Bradley
Prof. Watson
CSCI 297
21 November 2020

Appendix C
Feature importance rankings from Random Forest feature selection



Appendix D
Scaled Distribution Example



Appendix E
GitHub Repository Link: https://github.com/tuckert23/final_project