Supervised learning project

Machine learning, statistical learning, deep learning and artificial intelligence

Martina Corsini ID 944506

MSc Data science and economics
Università degli studi di Milano

July 2021

Abstract: The aim of this research is to understand which are the elements that mostly affect the instructional expenditure per student of the US colleges presented in the data. In order to perform this task, the research exploits two important supervised methods: the multiple linear regression and the regression tree. Furthermore, some more sophisticated models are implemented both in the linear regression framework and on the tree-based method, in order to make the analysis more complete and to reach higher performance. With regard to the multiple linear regression, a study of the multi-collinearity is done in order to ruled out predictors that are too correlated to each other; polynomial regression is implemented so to understand if some non-linear models could perform better on the data; model selection is applied exploiting a shrinkage method, the so called Lasso, in order to simplify the model and to consequently increase its predictive power. With respect to the tree-based methods, the basic regression tree is pruned in order to reduce overfitting on the dataset; furthermore, using the random forest technique, the predictive power of the model is increased.
Keywords: Multiple linear regression, polynomial regression, Lasso method, regression tree, pruning, random forest.

Contents

List of Figures

List of Tables

1 Introduction

While the United States remains a dream destination for students all over the world, it is also among the most expensive choices, with students spending an average of $99,417 over the course of their degree, covering a range of expenses including tuition fees, room and board costs, bills and personal expenses [1]. It could be therefore useful to understand which are the elements that mostly affect the instructional expenditure per student of the US colleges. In order to reach this goal, I took into consideration two different supervised methods: the multiple linear regression and the regression tree. Furthermore, I implemented some more sophisticated models both in the linear regression framework and on the tree-based method, in order to make the analysis more complete and to reach higher performance. With regard to the multiple linear regression, a study of the multi-collinearity was done in order to ruled out predictors that are too correlated to each other; polynomial regression was implemented so to understand if some non-linear models could perform better on the data; model selection was applied exploiting a shrinkage method, the so called Lasso, in order to simplify the model and to consequently increase its predictive power. With respect to the tree-based methods, the basic regression tree was pruned to reduce overfitting on the dataset; furthermore, using the random forest technique, the predictive power of the model was increased.

2 Theoretical background [2]

2.1 Linear regression

The first supervised method I used in my analysis is the linear regression. Linear regression models are parametric methods that aim to model the relationship between a quantitative target variable and one or more explanatory variables with linear predictor functions, by estimating the coefficients minimizing the sum of the squares of the distances between the observed data and the predicted ones, namely the residuals. The quality of a linear regression fit is typically assessed using the residual standard error (RSE), namely the average amount that response deviates from the true regression line, and the R^2 statistic, the proportion of variance explained.

2.1.1 Multiple linear regression

While the case of one explanatory variable is called simple linear regression, for more than one features the process is called multiple linear regression. This method gives each predictor a separate slope coefficient in a single model and it is considered a better approach compared with the fitting of separate simple linear regression model for each predictor. The values that minimizes the RSS (residual sum of squares) are the multiple least squares regression coefficient estimates. With the F-statistic, we can test the null hypothesis that all the coefficients are zero, so to be able to answer the question: Is there a relationship between the response and (at least one) predictor? When there is no relationship, F-statistic is expected to be close to 1. Note that it adjusts for the number of predictors. In order to assess the quality of the model fit, one could exploit the Multiple R^2. Howerver, this measure is not adjusted for the number of predictors, so it is convenient to use the Adjusted R^2.

2.1.2 Polynomial regression

One possible extension of the linear regression is the polynomial regression. Through the analysis of the residual plots, one can identify non-linearity of the data. Therefore, it could be useful to expand the linear model by adding extra predictors, obtained by raising each of the original predictors to a power, in order to produce a non-linear curve.

2.1.3 Subset selection

It is often the case that some of the variables used in multiple regression model are in fact not associated with the response. Therefore, we can proceed with the elimination of these variables. One possible alternative is the subset selection. The most intuitive approach is the best subset selection, in which we fit separate regression for each possible combination of the n predictors in order to identify the best subset. This approach is often computationally infeasible since it requires the test of an huge number of models. For this reason, forward and backward stepwise selection could be used. Forward stepwise begins with a model containing no predictors, and then adds

predictors to the model, one at the time. At each step, the variable that gives the greatest additional improvement to the fit is added to the model. Then, one model is selected. In the second approach, the starting model is the full one and at each step the least important variable is removed. Note that in both of this approaches is not guaranteed to obtain the best model.

2.1.4 Shrinkage methods

Model selection can be also performed using shrinkage methods on the standardized variables, through the use of constraints that shrink the coefficients of the regression towards zero, thus excluding them if their coefficients are exactly or very close to zero. The Ridge regression and the Lasso one are two of these techniques. They use a tuning parameter λ to shrink the coefficients towards zero. If the tuning parameter is 0, we have no regularization. To select the optimal λ, cross-valdiation can be used on a grid of λ values. Ridge and Lasso regression only differ in the fact that Ridge regression shrinks all the coefficients to a non-zero value (since the cost function is $RSS + \lambda \sum_{j=1}^{p} \beta_j^2$), so it will always include all the predictors in the final model, while the Lasso shrinks some of the coefficients all the way to zero (the Lasso coefficients minimize the quantity $RSS + \lambda \sum_{j=1}^{p} |\beta_j|$). In order to perform variable selection, I decided to use Lasso regression.

2.2 Tree-based methods

The second supervised method I used in my analysis is the regression tree. Tree-based models are non parametric method that use a series of if-then rules to divide the predictor space in a set of different subspaces, in order to generate predictions for each of the final leaves, by imposing that predictors that are in the same subspace have the same output value. All tree-based models can be used for either regression (predicting numerical values) or classification (predicting categorical values). The application of splitting rules starts at the top of the tree, so these approaches are called $top - down$. They are also *greedy*, because the best split at each step is made considering only that particular step.

2.2.1 Regression trees

Regression trees' building is composed of two steps. First of all, the predictor space is segmented into a number of distinct and non overlapping regions, usually with the form of boxes; then, predictions for each of the final regions (leaves) are generated using the mean of the training observations in the given region. There are essentially two key components to building a decision tree model: determining which features to split on and then deciding when to stop splitting. With regards to the first point, the aim is to split the observations at each step in a way that minimize the RSS. With respect to the second issue, usually the process is iterated, looking for the best predictor and the best cutpoint in order to split one of the previously identified regions, until each box contains no more than a fixed quantity of points. In order to avoid overfitting,

we can apply pruning: pruned trees are basically subtrees that we obtain by setting a value for the complexity parameter the gives us the lowest test error estimate for our model. To decide the tuning parameter, cross validation can be applied.

2.2.2 Bagging and random forest

Regression tree has the great advantage to be easy to interpret, but they are not competitive in terms of prediction accuracy, since they usually suffer from high variance. While pruning is a good method of improving the predictive performance of a decision tree model, a single decision tree model will not generally produce strong predictions alone. To improve our model's predictive power, we can build many trees and combine the predictions, since we know that averaging a set of observation reduces variance. One technique that can be used is the bagging: it builds many regression trees at a time by randomly sampling with replacement, or bootstrapping, from the original training set. Random forest is an improvement of this method, in which a random selection of m predictors is chosen for each split as candidates. Such procedure is done in order to avoid that the bagged trees are too correlated due to the fact that there only few predictor that dominates the others.

3 Dataset

The dataset considered is the College dataset from the R library ISLR, that is taken from the 1995 U.S. News World Report's Guide to America's Best Colleges. It contains data for 777 US Colleges. I selected several variables from the College dataset that could be linked with my target variable, namely the instructional expenditure per student. The following schema explains which variables I used in my research.

Variable	Description
Private	Private or public university
Apps	Applications received
Top10perc	Pct. new students from top 10 % of H.S. class
Top25perc	Pct. new students from top 25 % of H.S. class
Outstate	Out-of-state tuition
Room.Board	Room and board costs
Books	Book costs
Personal	Personal spending
Expend	Instructional expenditure per student (Target)

Table 1: Variables

Prior to go deeper into the evaluation of the statistical methods applied, I started an exploratory view of the data (Fig 1) by displaying the boxplot for each of the selected variables to get an overview of the main characteristics of the distributions.

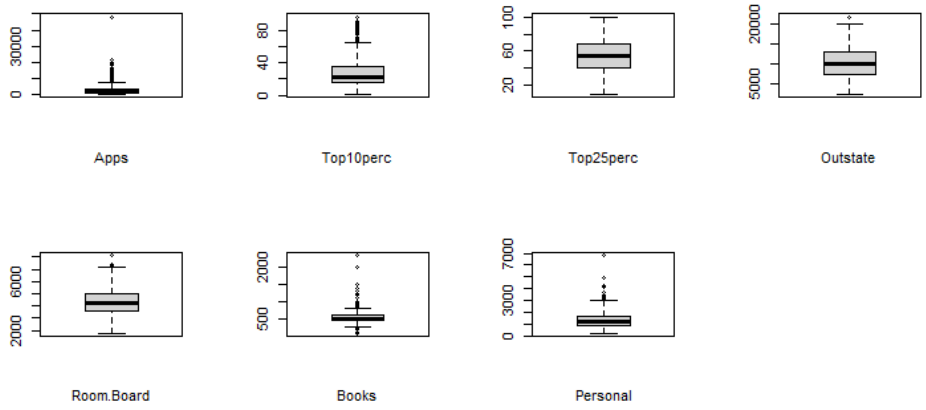

Figure 1: Boxplots of features

I also explored the distribution of the response variable. Since the variable is skewed (Fig 2), I procedeed with the normalization of the variable (Fig 3), so to be able to make the results easier to interpret.

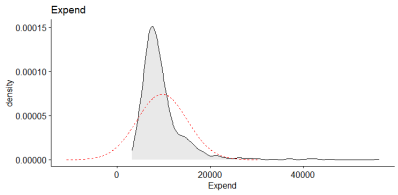

Figure 2: Target variable

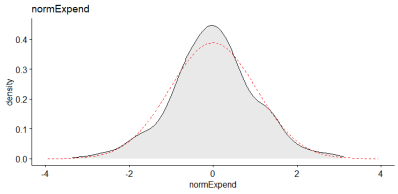

Figure 3: Normalized target variable

4 Findings

4.1 Multiple linear regression

As first step, I splitted the dataset into training and test sample set. The train-test split is useful to evaluate the performance of the algorithms applied. In the first model I included all of the predictors. Being the F-statistic equal to 92.85 (well-above 1), as it is shown in Fig 4, I could infer that there is an effective relation between at least one of the predictors and the response variable. The Adjusted R^2 value of 0.655 indicates that the model is capturing a good level of variance of our dataset (the closer to 1 the better the fit).

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.538e+00  1.949e-01 -13.020  < 2e-16 ***
PrivateYes  -1.571e-02  1.097e-01  -0.143  0.88626
Apps         7.698e-06  1.386e-05   0.555  0.57891
Top10perc    2.076e-02  3.887e-03   5.340  1.6e-07 ***
Top25perc   -5.082e-03  3.291e-03  -1.544  0.12331
Outstate     1.345e-04  1.325e-05  10.150  < 2e-16 ***
Room.Board   1.186e-04  3.691e-05   3.212  0.00143 **
Books        2.421e-04  1.970e-04   1.229  0.21988
Personal     9.952e-05  5.112e-05   1.947  0.05232 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6096 on 379 degrees of freedom
Multiple R-squared:  0.6621,    Adjusted R-squared:  0.655
F-statistic: 92.85 on 8 and 379 DF,  p-value: < 2.2e-16
```

Figure 4: Full model

In the second fit of the model (Fig 7), I excluded those predictors which were too correlated to each other. Having collinearity between predictors may imply upward or downward bias in the regression coefficients estimates. In order to understand the multicollinearity, I used the VIF (variance inflation factor), that gives us a measure of how much the variance of a regression coefficient is inflated due to multicollinearity in the model. It could be also useful to use the correlation matrix of Fig 6 in order to understand which variable are more correlated. As expected, Top10perc and Top25perc show an high level of correlation (Fig 5), so I decided to remove the latter.

```
> vif(full.model) # variance inflation factors
   Private       Apps  Top10perc  Top25perc   Outstate Room.Board      Books   Personal
  2.540386   2.117547   5.335416   4.601355   3.074049   1.747745   1.073549   1.156729
> sqrt(vif(full.model)) > 2 # problem?
   Private       Apps  Top10perc  Top25perc   Outstate Room.Board      Books   Personal
     FALSE      FALSE       TRUE       TRUE      FALSE      FALSE      FALSE      FALSE
>
```

Figure 5: VIF

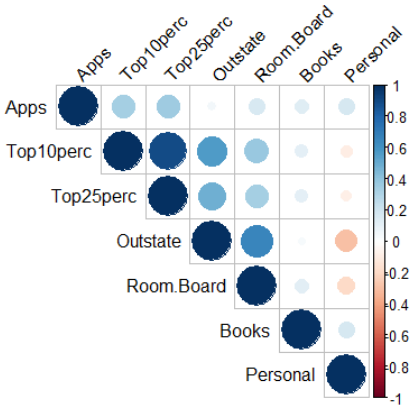

Figure 6: Correlation matrix

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.675e+00  1.736e-01 -15.408  < 2e-16 ***
PrivateYes  -8.179e-04  1.095e-01  -0.007  0.99405
Apps         7.641e-06  1.388e-05   0.550  0.58241
Top10perc    1.601e-02  2.386e-03   6.710 7.07e-11 ***
Outstate     1.331e-04  1.324e-05  10.049  < 2e-16 ***
Room.Board   1.173e-04  3.697e-05   3.171  0.00164 **
Books        2.310e-04  1.972e-04   1.171  0.24226
Personal     1.006e-04  5.121e-05   1.965  0.05016 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6107 on 380 degrees of freedom
Multiple R-squared:   0.66,     Adjusted R-squared:  0.6538
F-statistic: 105.4 on 7 and 380 DF,  p-value: < 2.2e-16

> |
```

Figure 7: Full model without collinearity

Finally, by looking at the p-value we can say that the lower the p-value the more significant will be the variable. In the final fit of the model (Fig 8), I excluded those predictors which were not significant. By removing the least significant predictor, I obtained an F-Statistic of 241.3, thus at the expense of a slightly lower Adjusted R-squared of 0.6507.

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.379e+00  1.263e-01 -18.828  < 2e-16 ***
Top10perc    1.773e-02  2.092e-03   8.476 5.08e-16 ***
Outstate     1.241e-04  1.090e-05  11.381  < 2e-16 ***
Room.Board   1.253e-04  3.591e-05   3.488 0.000542 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6133 on 384 degrees of freedom
Multiple R-squared:  0.6534,    Adjusted R-squared:  0.6507
F-statistic: 241.3 on 3 and 384 DF,  p-value: < 2.2e-16
```

Figure 8: Full model with significant features

Looking at the final model, we can infer that the elements that mostly affect the instructional expenditure per student of the US colleges are three: the cost of board and accommodation, the cost of tuition for the out-of-state students and the percentage of new students from top 10 % of high schools, that could be interpreted as an indicator of the prestige of the College. Other predictors are instead not significant, such as the public or private nature of a college, the number of application received by the school, the personal expenses and the expenses for books. It may seem surprising that the private or public nature of a University does not impact the instructional expenditure for student. However, the reason lies in the nature of the college system in the US. In fact, public universities in the US have two tuition fee rates: one for state residents and one, that is very expensive, for everyone else. The average expenditure in tuition for public college is therefore an average of in-state tuition and out-of-state tuition. On the other hand, private universities tend to have a more diverse student population (both from different states and different countries) precisely due to the fact that tuition is the same price for all students.

After having selected the predictors, I evaluated the model on the test set. In the plots of Fig 9 and Fig 10, I meant to compare performance of the final model in the train set and test set respectively. As it appears from the two graphs, there is no significant difference between the two; also, the MSE for the train set is pretty close to the MSE for the test set.

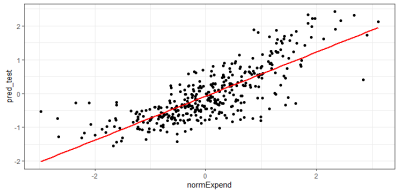

Figure 9: Test

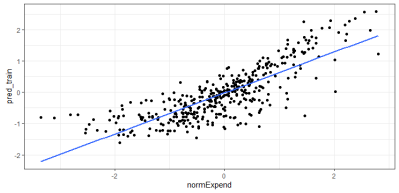

Figure 10: Train

A further analysis that could be done is about the possibility of some kind of non-linearity in the data, which could be better approximated by polynomial models. As we can see from the Fig 11 and from the absence of any trend in the pattern of the residuals of the linear model (Fig 12), it does not seem like there is some specific path that we cannot catch by using a linear model.

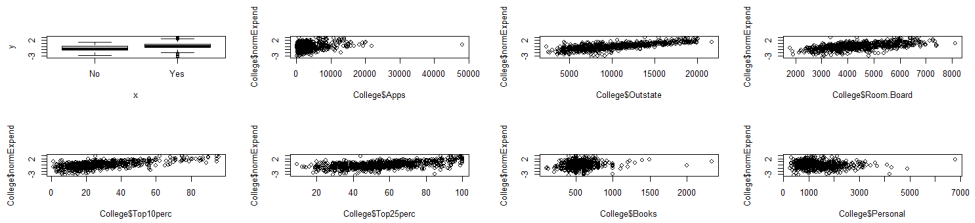

Figure 11: Plot of features against target

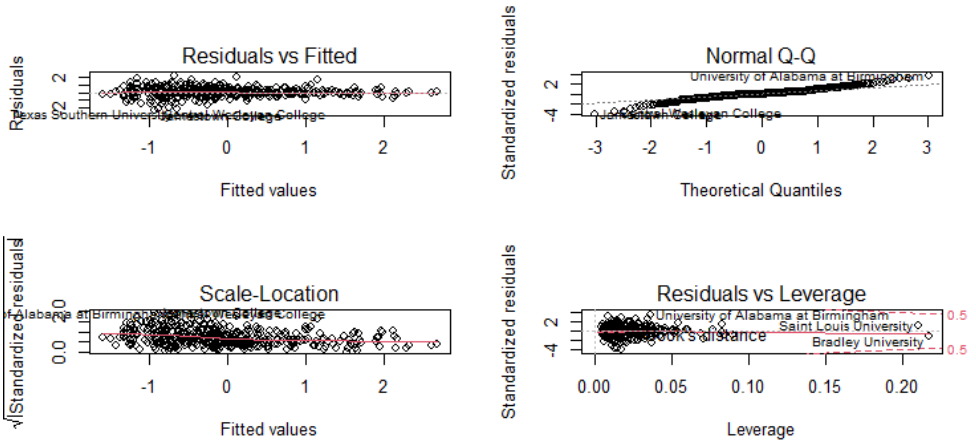

Figure 12: Residuals

Nevertheless, I proceeded with the implementation of polynomial models on 'Top10perc' to check which is the optimal degree of flexibility according to the parsimonious modeling principle, which takes into account the bias-variance trade off. Using the k-fold cross validation method on a range of 10 degrees of polynomiality, I found out that the linear regression gives the lowest MSE and using the ANOVA table (Fig 13) I rejected the null hypothesis that linear and quadratic regression are different. In addition, also the Stepwise Selection method on the cubic regression gave me the same results (Fig 14), since, despite the fact that all the variables are included in the model, we can observe that the coefficients for the quadratic and cubic variables are not significant.

```
> # ANOVA
> anova(lin, pol_mod2)
Analysis of Variance Table

Model 1: normExpend ~ Top10perc + Outstate + Room.Board
Model 2: normExpend ~ poly(Top10perc, 2) + Outstate + Room.Board
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    384 144.46
2    383 144.17  1   0.28547 0.7584 0.3844
```

Figure 13: ANOVA

```
Residuals:
     Min       1Q   Median       3Q      Max
-2.55550 -0.27141 -0.00549  0.25762  2.22701

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)        -1.091e+00  1.653e-01  -6.597 1.41e-10 ***
Top10perc           1.777e-02  2.179e-03   8.156 5.03e-15 ***
poly(Outstate, 3)1  1.000e+01  8.845e-01  11.307  < 2e-16 ***
poly(Outstate, 3)2 -1.934e-02  6.411e-01  -0.030 0.975951
poly(Outstate, 3)3  1.687e-01  6.155e-01   0.274 0.784155
Room.Board          1.253e-04  3.626e-05   3.455 0.000613 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6149 on 382 degrees of freedom
Multiple R-squared:  0.6535,    Adjusted R-squared:  0.649
F-statistic: 144.1 on 5 and 382 DF,  p-value: < 2.2e-16

> par(mfrow=c(2,2))
> plot(step)
>
```

Figure 14: Stepwise selection

Finally, I applied a shrinkage method, the Lasso one, in order to check if the model could be simplified. As we said, unlike Ridge regression, the Lasso does involve best subset selection. First of all, I performed a 10-fold cross-validation in order to select the value of the penalty parameter that minimize the MSE in the model. The resulting value associated to the smallest cross-validation error is 0.1. Therefore, we are not far to the linear model. Furthermore, the MSE associated with the regression model is slightly higher than the MSE in the final linear model. The variables that are equal to 0 are the least significant variables that I already excluded in the final fit of the multiple regression model, as we can see in Fig 15.

```
> ## Final model with Lasso Regression
> out=glmnet(x,y,alpha=1,lambda=grid)
> lasso.coef= predict(out, type="coefficients",s=bestlam)[1:9,]
> as.table(lasso.coef)
  (Intercept)       Private          Apps     Top10perc     Top25perc      Outstate
-1.892744e+00  0.000000e+00  2.040628e-06  1.327634e-02  0.000000e+00  1.162201e-04
   Room.Board         Books      Personal
 7.052791e-05  0.000000e+00  0.000000e+00
```

Figure 15: Lasso regression

4.2 Tree-based methods

In the second section of my analysis, I considered the tree based methods. First of all, I performed a basic regression tree on the training set (Fig 16). The variables actually used to construct the tree are 5, with 'Outstate' that seems the most important predictor since it is used in multiple levels and even at the root one.

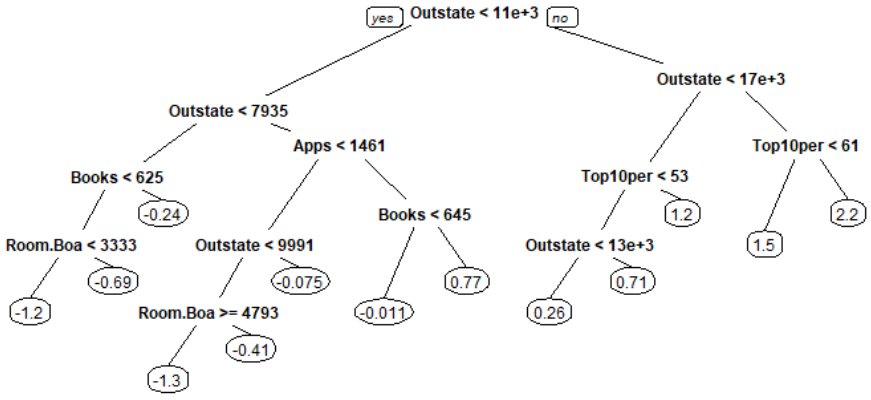

Figure 16: Regression tree

However, this result suffers from high variability and low bias. Therefore, I pruned the tree (Fig 18) by setting a value for the complexity parameter *cp* the gives the lowest test error estimate for the model. According to the Fig 17, I chose cp= 0.039. Note that in this case, as expected, the model resulted in a greater MSE since the pruned model is more generalizable, but on the other hand overfitting to the training data was reduced.

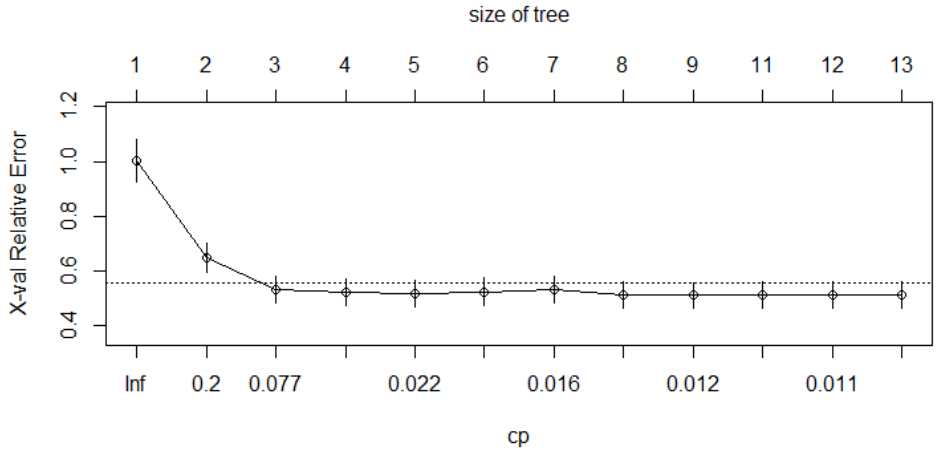

Figure 17: Complexity parameter

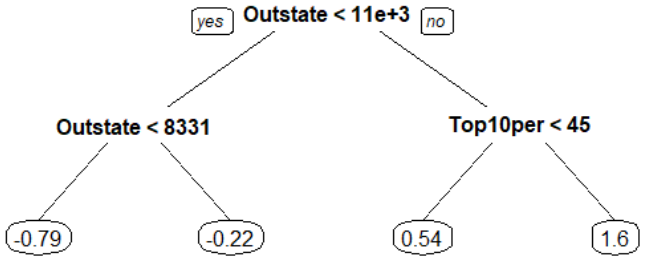

Figure 18: Pruned tree

Finally, I performed random forest, so to add randomization and consequently reduce the variance. I chose to perform Random Forest instead of Bagging because of the presence of a strong predictor in the dataset, which could lead to build very similar trees for each sub-sample, thus generating less reduction in the variance. I set the number of predictors for each split equal to 3, since this is the number of predictors that I found to be significant after the linear regression analysis. Since it was no longer possible to visualize the results using a single tree, I computed the variable importance measure to find the most relevant predictors (Fig 19). Again, the most important features in the prediction of the instructional expenditure per student of the US colleges are the cost of tuition for the out-of-state students, the cost of board and accommodation and the percentage of new students from top 10 % of high schools.

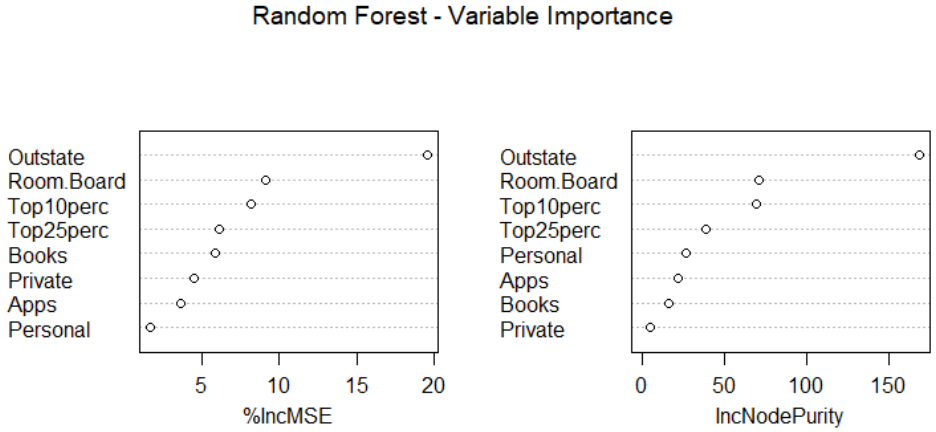

Figure 19: Variable importance measure

5 Conclusions

The aim of this research was to understand which were the elements that mostly affected the instructional expenditure per student of different US colleges.The variables that I used in my analysis were retrieved by the College dataset from the R library ISLR. I decided to select as variables that could be linked with my target variable, namely the instructional expenditure per student, the following features: the public or private nature of the College; the number of applications received by the College; the number of new students from the top 10 % of the High school class; the number of new students from the top 25 % of the High school class; the Out-of-state tuition; the average costs for board and room; the average costs of books; the average personal expenditure.
In order to reach my goal, I took into consideration two different supervised methods: the multiple linear regression and the regression tree.
Speaking about multiple linear regression, I developed three possible models: in the first one, I included all of the predictors; in the second fit of the model, I excluded the Top25perc variable since it showed a problem of multi-collinearity with the Top10perc variable; in the final fit of the model, I excluded those predictors which were not significant looking at the p-value. The last model showed an huge F-statistic and a good level of Adjusted R^2. Furtheremore, it did not overfit, as I could infer looking at the values of MSE for the test and the training set. Looking at the final model, I could infer that the elements that mostly affect the instructional expenditure per student of the US colleges were three: the cost of board and accommodation, the cost of tuition for the out-of-state students and the percentage of new students from top 10 % of high schools, that could be interpreted as an indicator of the prestige of the College. The fact that the private or public nature of a University does not impact the instructional expenditure for student was explained considering the nature of the college system in the US, for which the public universities have two differet tuition fee rates for state residents and for everyone else while the private universities have the same tuition for all students.
The absence of any trend in the pattern of the residuals of the linear model showed that polynomial models were not needed. Furthermore, I corroborated this hypothesis using the k-fold cross validation method on a range of 10 degrees of polynomiality and the stepwise selection method on the cubic regression.
Finally, the application of the Lasso assigned non-zero values to the same variables that I already included in the final fit of the multiple regression model.
In the second section of my analysis, I considered the tree based methods. The variables actually used to construct the basic regression tree were 5, with 'Outstate' that seemed the most important predictor since it is used in multiple levels and even at the root one. Pruning the tree, only two variables were considered. Finally, I performed random forest, so to add randomization and consequently reduce the variance. Again, the most important features in the prediction of the instructional expenditure per student of the US colleges were the cost of tuition for the out-of-state students, the cost of board and accommodation and the percentage of new students from top 10 % of high schools

6 R codes

```r
#call libraries
library (ISLR)
library(ggpubr)
library(bestNormalize)
library(e1071)
library(Hmisc)
library(car)
library(boot)
library(corrplot)
library(MASS)
library(glmnet)
library(tree)
library(rpart)
library(rpart.plot)
library(randomForest)
library(caTools)

#choose dataset
fix(College)
names(College)
College= na.omit(College)
c = c(1,2,5,6,9,10,11,12,17)
College= College [,c]

#exploratory analysis
par(mfrow=c(2,4))
for (i in c(2,3,4,5,6,7,8)){
  boxplot(College[,i], data = College, xlab=colnames(College)[i])
}
summary(College$Expend)
ggdensity(College, x = "Expend", fill = "lightgray", title = "Expend") +
  stat_overlay_normal_density(color = "red", linetype = "dashed")

#pick the best normalization process automatically
(BNobject <- bestNormalize(College$Expend))
par(mfrow = c(1,2))
MASS::truehist(BNobject$x.t,
               main = paste("Best Transformation:",
                            class(BNobject$chosen_transform)[1]), nbins =
    12)
College$normExpend <-BNobject$x.t
skewness(College$normExpend, na.rm = TRUE)
ggdensity(College, x = "normExpend", fill = "lightgray", title = "
    normExpend") +
  stat_overlay_normal_density(color = "red", linetype = "dashed")

#splitting between train set and test set
set.seed(123)
row.number = sample(1:nrow(College),0.5*nrow(College))
train= College[row.number,]
test= College[-row.number,]

#fit the multiple linear regression model
```

```r
College$Private <- ifelse(College$Private == 'Yes', 1, 0)
full.model <- lm(normExpend ~ .-Expend, data = train)
summary(full.model)
par(mfrow=c(2,2))
plot(full.model)
plot(hatvalues(full.model))

#check for collinearity issue:
vif(full.model)
sqrt(vif(full.model)) > 2

#correlation matrix
a<-c(2,3,4,5,6,7,8)
res<-cor(College[a])
round(res, 2)
symnum(res, abbr.colnames = FALSE)
rcorr(as.matrix(College[a]))
flattenCorrMatrix <- function(cormat, pmat) {
  ut <- upper.tri(cormat)
  data.frame(
    row = rownames(cormat)[row(cormat)[ut]],
    column = rownames(cormat)[col(cormat)[ut]],
    cor  =(cormat)[ut],
    p = pmat[ut]
  )
}
res1<-rcorr(as.matrix(College[a]))
flattenCorrMatrix(res1$r, res1$P)
col<- colorRampPalette(c("blue", "white", "red"))(20)
heatmap(x = res, col = col, symm = TRUE, Colv = NA, Rowv = NA)
corrplot(res, type = "upper",
         tl.col = "black", tl.srt = 45)

#fit the multiple linear regression model without collinearity
lm.fit <- lm(normExpend ~ Private+ Apps+Top10perc+ Outstate+ Room.Board+
    Books+Personal, data = train)
summary(lm.fit)

#fit the multiple linear regression model without useless variables
lm.fit1 <- lm(normExpend ~ Top10perc+ Outstate+ Room.Board, data = train)
summary(lm.fit1)

#prediction
pred_test = predict(lm.fit1, newdata = test)
mse = sum((pred_test - test$normExpend)^2)/length(test$normExpend)
c(MSE_test = mse, R2=summary(lm.fit1)$r.squared)
pred_train= predict(lm.fit1, newdata=train)
mse= sum((pred_train-train$normExpend)^2)/length(train$normExpend)
c(MSE_train=mse, R2=summary(lm.fit1)$r.squared)
par(mfrow=c(1,1))
ggplot(test, aes(x = normExpend, y = pred_test))+
  geom_point() +
  geom_smooth(method = 'lm', se = FALSE, col= 'red') +
  theme_bw()       # Plotting of the predicted response variable using
    test set
ggplot(train, aes(x = normExpend, y = pred_train)) +
```

```
  geom_point() +
  geom_smooth(method = 'lm', se = FALSE) +
  theme_bw()      # Plotting of the predicted response variable using
    training set

#predictors plotting
par(mfrow=c(2,2))
plot(College$Private,College$normExpend)
plot(College$Apps,College$normExpend)
plot(College$Top10perc,College$normExpend)
plot(College$Top25perc,College$normExpend)
plot(College$Outstate,College$normExpend)
plot(College$Room.Board,College$normExpend)
plot(College$Books,College$normExpend)
plot(College$Personal,College$normExpend)

#polynomial regression
plot (College$normExpend, College$Top10perc)
lin <- lm(normExpend ~ Top10perc+ Outstate+ Room.Board, data = train)
pred_test = predict(lin, newdata = test)
mse = sum((pred_test - test$normExpend)^2)/length(test$normExpend)
c(MSE_test = mse, R2=summary(lin)$r.squared)
pol_mod2 <- lm(normExpend ~poly(Top10perc, 2)+ Outstate + Room.Board,
    data = train)
pred_test = predict(pol_mod2, newdata = test)
mse = sum((pred_test - test$normExpend)^2)/length(test$normExpend)
c(MSE_test = mse, R2=summary(pol_mod2)$r.squared)
pol_mod3 <- lm(normExpend ~poly(Top10perc,3)+ Outstate + Room.Board, data
     = train)
pred_test = predict(pol_mod3, newdata = test)
mse = sum((pred_test - test$normExpend)^2)/length(test$normExpend)
c(MSE_test = mse, R2=summary(pol_mod3)$r.squared)

#k fold cross validation
set.seed(100)
cv.error.10 = rep(0,10)
for (i in 1:10){
  glm.fit = glm (normExpend ~poly(Top10perc,i )+ Outstate + Room.Board,
    data = College)
  cv.error.10[i]= cv.glm (College,glm.fit, K= 10)$delta[1]
}
cv.error.10

#ANOVA
anova(full.model, pol_mod2)

#stepwise selection
step <- stepAIC(pol_mod3, direction="both")
step$anova # display results
summary(step)
par(mfrow=c(2,2))
plot(step)

#shrinkage methods
#ridge Regression (alpha = 0)
x= model.matrix(normExpend~.-Expend,College)[,-1]
```

```
y= College$normExpend
set.seed(333)
train=sample(1:nrow(x),nrow(x)/2)
test= (-train)
y.test=y[test]
# CV for tuning parameter lambda
set.seed(111)
cv.out= cv.glmnet(x[train,],y[train],alpha=0,standardize=TRUE)
plot(cv.out,col= 'red')
bestlam=cv.out$lambda.min
c(`Best Lambda`= bestlam)
# MSE of Ridge Regression using optimal lambda
grid= 10^seq(10,-2,length=100)
ridge.mod= glmnet(x[train,],y[train],alpha=0,
                  lambda=grid,thresh=1e-12,standardized= TRUE)
ridge.pred= predict(ridge.mod, s=bestlam, newx=x[test,])
c(MSE= mean((ridge.pred-y.test)^2))
out= glmnet(x,y,alpha=0)
ridge.coef= predict(out,type = "coefficients", s=bestlam)[1:9,]
as.table(ridge.coef)
plot(ridge.mod,xvar='lambda',label=TRUE)
title(main= 'Ridge', line = 2.4)

#lasso regression (alpha=1)
# CV for tuning lambda parameter for lasso regression
set.seed(121)
cv.lasso=cv.glmnet(x[train,],y[train],alpha=1)
plot(cv.lasso,col='red')
bestlam=cv.lasso$lambda.1se
c(`Best Lambda Lasso`=bestlam)
lasso.mod= glmnet(x[train,],y[train],alpha= 1,
                  lambda=grid,thresh=1e-12,standardized= TRUE)
lasso.pred=predict(lasso.mod,s=bestlam,newx=x[test,])
c(`MSE Lasso`=mean((lasso.pred-y.test)^2))

#final model with Lasso Regression
out=glmnet(x,y,alpha=1,lambda=grid)
lasso.coef= predict(out, type="coefficients",s=bestlam)[1:9,]
as.table(lasso.coef)
plot(lasso.mod,xvar='lambda',label=TRUE)
title(main= 'Lasso', line = 2.4)

#tree-Based Methods
#regression tree
set.seed(121)
split = sample.split(College$normExpend, SplitRatio = 0.5)
train = subset(College, split==TRUE, standardized= TRUE)
test = subset(College, split==FALSE, standardized = TRUE)
tree.dataset<- tree(normExpend ~.-Expend, data=train)
plot (tree.dataset)
text(tree.dataset, pretty= 0, digits=3, cex = 0.75)
title(main='Regression Tree')
tree.r = rpart(normExpend~.-Expend, data=train)
prp(tree.r)
summary(tree.r)
printcp(tree.r)
```

```
plotcp(tree.r)

#accuracy of base model regression tree (MSE and RMSE)
tree.pred= predict(tree.r, newdata=test)
tree.mse= mean((tree.pred-test$normExpend)^2)
tree.rmse= sqrt(mean((tree.pred-test$normExpend)^2))
c(MSE= tree.mse)
c(RMSE= tree.rmse)

#pruning tree with cp optimal
tree.pruned = prune(tree.r, cp= 0.07)
prp(tree.pruned)

#accuracy of the pruned regression tree
pred.pruned = predict(tree.pruned, test)
pruned.mse= mean((pred.pruned-test$normExpend)^2)
pruned.rmse= sqrt(mean((pred.pruned-test$normExpend)^2))
c(Pr.MSE= pruned.mse)
c(Pr.RMSE= pruned.rmse)

#random forest
rf_tree <- randomForest(normExpend ~.-Expend, data=train,
                        mtry=3, importance = TRUE, ntree=100)

#accuracy of the random forest
pred.rf = predict(rf_tree , newdata=test)
mse.rf= mean((pred.rf-test$normExpend)^2)
c('MSE random forest' = mse.rf)

#variable importance measure
importance(rf_tree)
varImpPlot(rf_tree, main = 'Random Forest - Variable Importance')
```

Listing 1: R Codes

References

[1] Reproduced with permission from The Value of Education, The price of success, published in 2018 by HSBC Holdings plc., London

[2] This chapter is based on the book "An introduction to statistical learning: with application in R" (2013), James G., Witten D., Hatie T.,Tibshirani R., Springer Reproduced with permission from The Value of Education, The price of success, published in 2018 by HSBC Holdings plc.'