DATA311 Project

Parsa Rajabi, Chelsey Hvingelby, Mackenzie Salloum, Cameron Chong, Jeff Bulmer

2019-04-04

The following libraries are required in order to run the markdown script properly. They can be installed from the CRAN repositories.

```
#install.packages("FNN")
#install.packages("mvtnorm")
#install.packages("mclust")
#install.packages("cluster")
#install.packages("tree")
#install.packages("randomForest")
#install.packages("fpc")
#install.packages("boot")
#install.packages("MASS")
library(FNN)
library(mvtnorm)
library(mclust)
```

```
## Package 'mclust' version 5.4.3
## Type 'citation("mclust")' for citing this R package in publications.
```

```
library(cluster)
library(fpc)
library(boot)
library(tree)
library(MASS)
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

The data in question is 500 observations of graduate admission students for Universities in India. It consists of categorical and continuous variables.

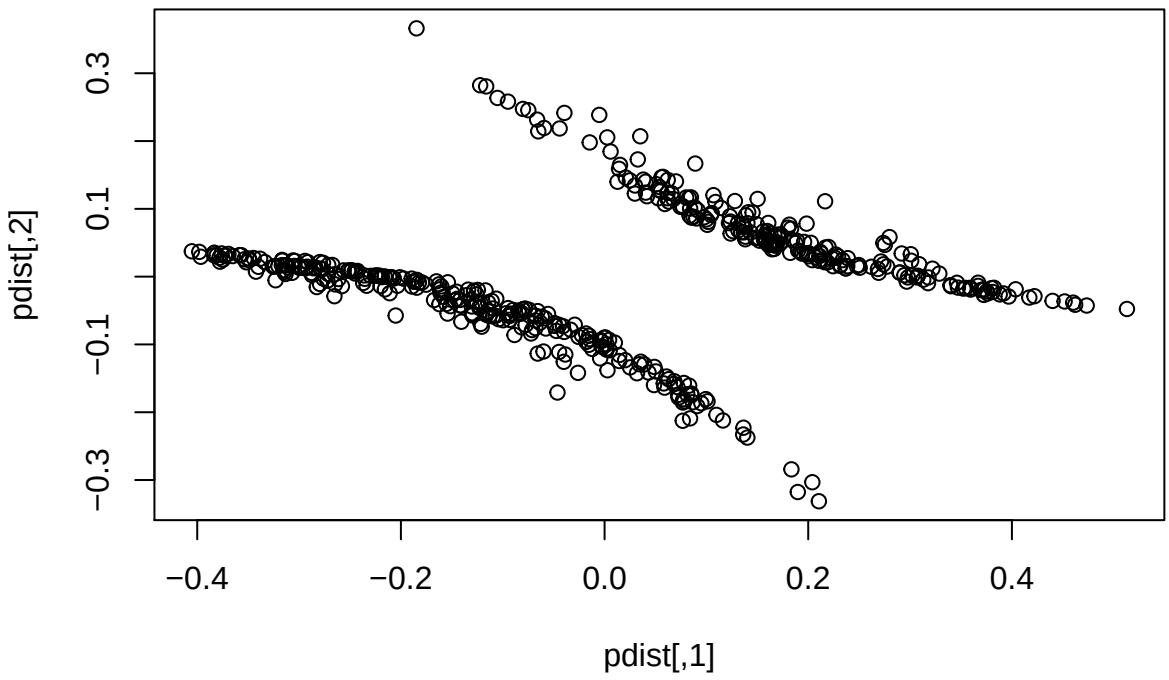
The dataset being explored in this report consists of graduate admissions data for students in India. Data was collected from 500 prospective graduate students, including various scores achieved in the Test of English as a First Language (TOEFL.Score) Graduate Record Examinations (GRE.Score), and scores indicating the strength of each candidates Statement of Purpose (SOP) and Letter of Recommendation (LOR). Other attributes include Undergraduate Cumulative GPA (CGPA), a unique identifier (Serial.No.), and whether or not the prospective student had Research Experience (Research). Finally, each candidate was polled about their confidence of being accepted into graduate school (Chance.of.Admit).

The data must be attached in order to run the analysis. As long as the file is in the same directory as the Rmd file it will run.

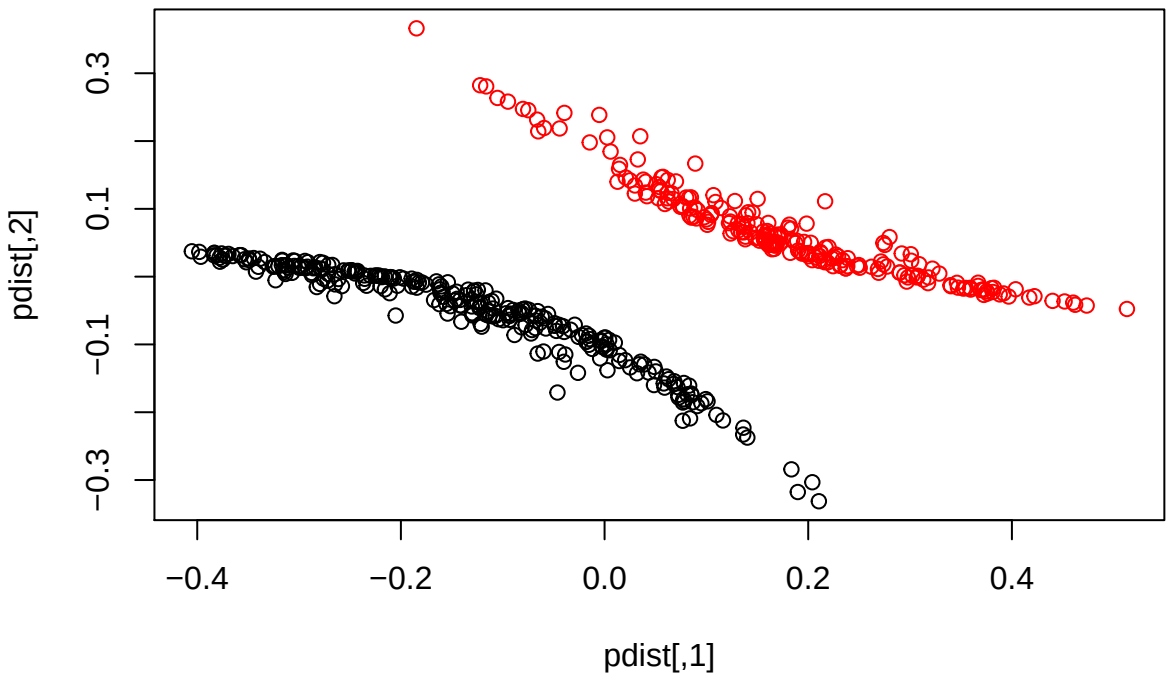
Clustering

We begin by computing the respective pairwise distances in our data, and plotting the output.

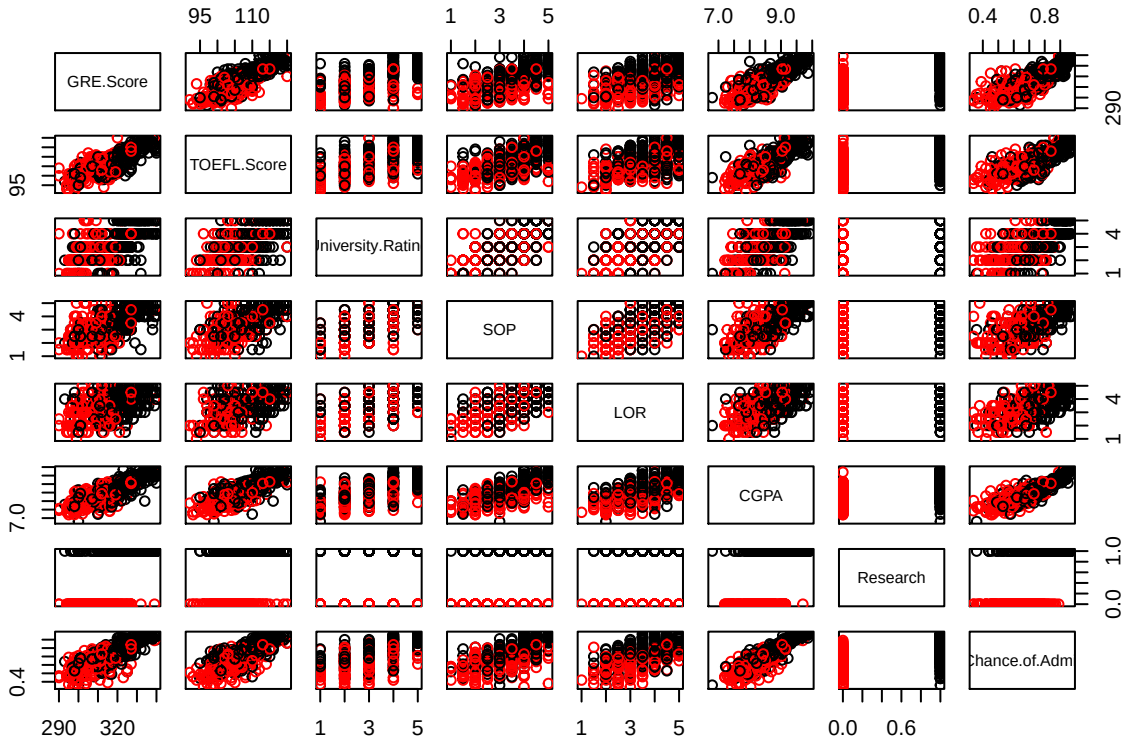
```
## Warning in daisy(admissionsData[, -1], metric = "gower"): binary
## variable(s) 7 treated as interval scaled
```



We quickly see that two clear groups appear. We can isolate these two groups using hierarchical clustering with single-linkage chaining.



We can then use scatterplots to show the entirety of the data, while still keeping the groups intact, to see if we can determine which predictors most affect these clusters.



We notice that, using the single linkage chaining from above, we can predict whether or not a student performs research almost perfectly.

So, by applying Gower's Distance on all predictors and using single-linkage chaining, we have two clear clusters directly coinciding with the presence of a research variable. This tells us that we should use Research as a response variable in models, in addition to Chance of Admit.

We can now perform analyses on the data to attempt to predict a candidate's Chance of Admission, as well as the presence of Research Experience.

Linear Models

PARSAS CODE GOES HERE

Bootstrap

JEFFS CODE HERE

Trees

We will apply 70/30 split of training and testing data. There are 500 observations, so we will have 350 training observations and 150 testing points.

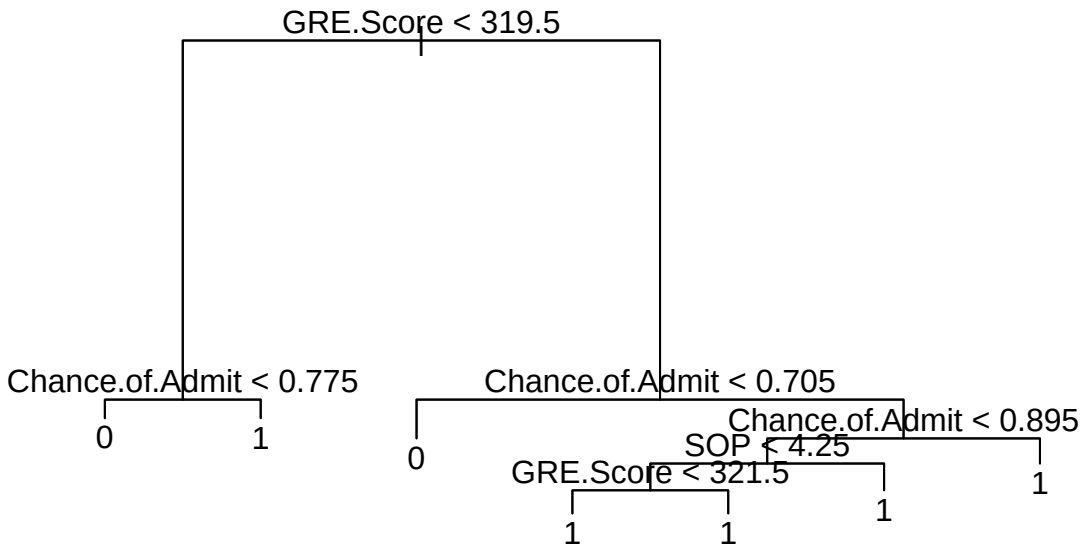
```
admissionsTreeData <- admissionsData[,-1]
trainindex <- sample(1:nrow(admissionsTreeData), 350)
```

```
admissionsTrain <- admissionsTreeData[trainindex, ]
admissionsTest <- admissionsTreeData[-trainindex, ]
```

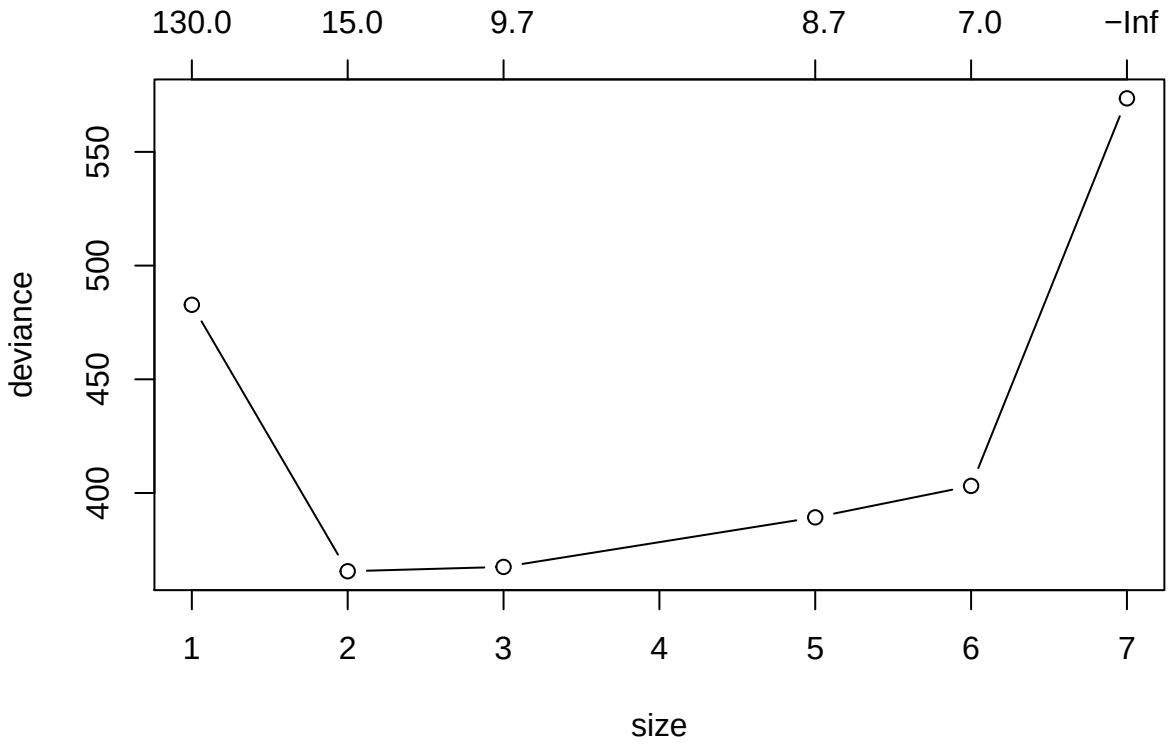
Research Tree

Cross Validation

```
set.seed(1232343124)
researchTree <- tree(as.factor(Research)~., data = admissionsTrain)
plot(researchTree)
text(researchTree, pretty=0)
```




```
researchTreeCV <- cv.tree(researchTree, FUN = prune.tree, K = 5)
plot(researchTreeCV, type = "b")
```




```
which.min(researchTreeCV$dev)
```

```
## [1] 5
```

```
researchTreeCV$dev
```

```
## [1] 573.5424 403.1185 389.2899 367.4901 365.5951 482.7631
```

```
researchTreeCV$dev
```

```
## [1] 573.5424 403.1185 389.2899 367.4901 365.5951 482.7631
```

```
researchTreeCV$size
```

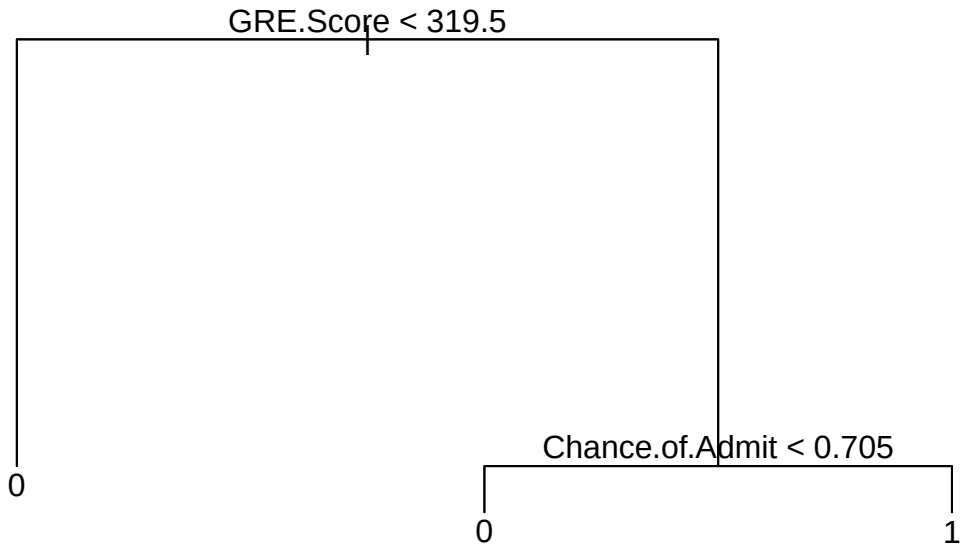
```
## [1] 7 6 5 3 2 1
```

```
which.min(researchTreeCV$dev)
```

```
## [1] 5
```

Cross Validation Suggests 3 terminal nodes would be best. So we will prune our tree to 3 terminal nodes

```
pruneResearchTreeCV <- prune.tree(researchTree, best=3)
plot(pruneResearchTreeCV)
text(pruneResearchTreeCV, pretty = 0)
```




```
summary(pruneResearchTreeCV)
```

```
##
## Classification tree:
## snip.tree(tree = researchTree, nodes = c(2L, 7L))
## Variables actually used in tree construction:
## [1] "GRE.Score"       "Chance.of.Admit"
## Number of terminal nodes:  3
## Residual mean deviance:  0.9544 = 331.2 / 347
## Misclassification error rate: 0.2143 = 75 / 350
```

Random Forest