Tree Methods Project

Guan-Yuan Wang

2020/9/3

Tree Methods Project

For this project we will be exploring the use of tree methods to classify schools as Private or Public based off their features.

Let's start by getting the data which is included in the ISLR library, the College data frame.

A data frame with 777 observations on the following 18 variables.

- Private A factor with levels No and Yes indicating private or public university
- Apps Number of applications received
- Accept Number of applications accepted
- Enroll Number of new students enrolled
- Top10perc Pct. new students from top 10% of H.S. class
- Top25perc Pct. new students from top 25% of H.S. class
- F.Undergrad Number of fulltime undergraduates
- P.Undergrad Number of parttime undergraduates
- Outstate Out-of-state tuition
- Room.Board Room and board costs
- Books Estimated book costs
- Personal Estimated personal spending
- PhD Pct. of faculty with Ph.D.'s
- Terminal Pct. of faculty with terminal degree
- S.F.Ratio Student/faculty ratio
- perc.alumni Pct. alumni who donate
- Expend Instructional expenditure per student
- Grad.Rate Graduation rate

Get the Data

Call the ISLR library and check the head of College (a built-in data frame with ISLR, use `data()` to check this.) Then reassign College to a dataframe called df

```r
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.6.3
```

```r
#data()
head(College)
```

```
##                              Private Apps Accept Enroll Top10perc Top25perc
## Abilene Christian University     Yes 1660   1232    721        23        52
## Adelphi University               Yes 2186   1924    512        16        29
## Adrian College                   Yes 1428   1097    336        22        50
## Agnes Scott College              Yes  417    349    137        60        89
## Alaska Pacific University        Yes  193    146     55        16        44
## Albertson College                Yes  587    479    158        38        62
##                              F.Undergrad P.Undergrad Outstate Room.Board Books
## Abilene Christian University        2885         537     7440       3300   450
## Adelphi University                  2683        1227    12280       6450   750
## Adrian College                      1036          99    11250       3750   400
## Agnes Scott College                  510          63    12960       5450   450
## Alaska Pacific University            249         869     7560       4120   800
## Albertson College                    678          41    13500       3335   500
##                              Personal PhD Terminal S.F.Ratio perc.alumni Expend
## Abilene Christian University     2200  70       78      18.1          12   7041
## Adelphi University               1500  29       30      12.2          16  10527
## Adrian College                   1165  53       66      12.9          30   8735
## Agnes Scott College               875  92       97       7.7          37  19016
## Alaska Pacific University        1500  76       72      11.9           2  10922
## Albertson College                 675  67       73       9.4          11   9727
##                              Grad.Rate
## Abilene Christian University        60
## Adelphi University                  56
## Adrian College                      54
## Agnes Scott College                 59
## Alaska Pacific University           15
## Albertson College                   55
```

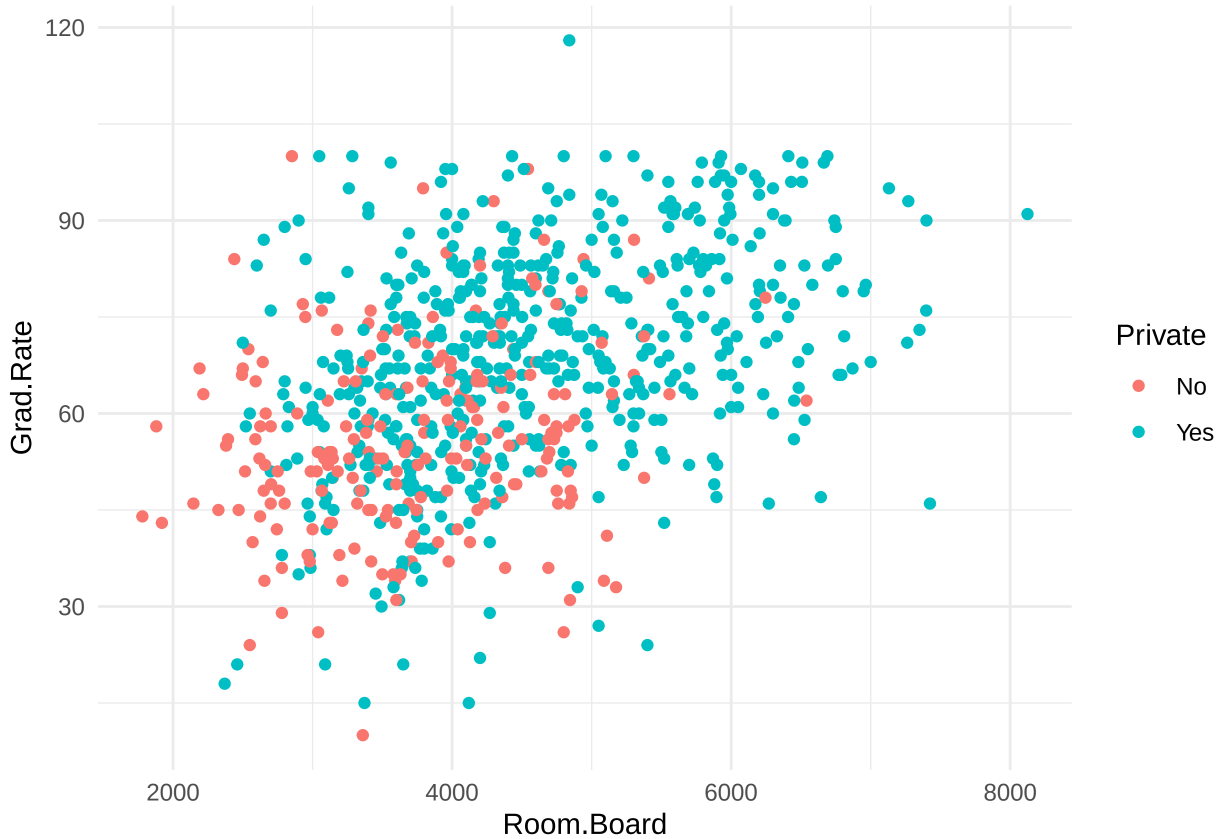
```r
df <- College
```

EDA

Let's explore the data!

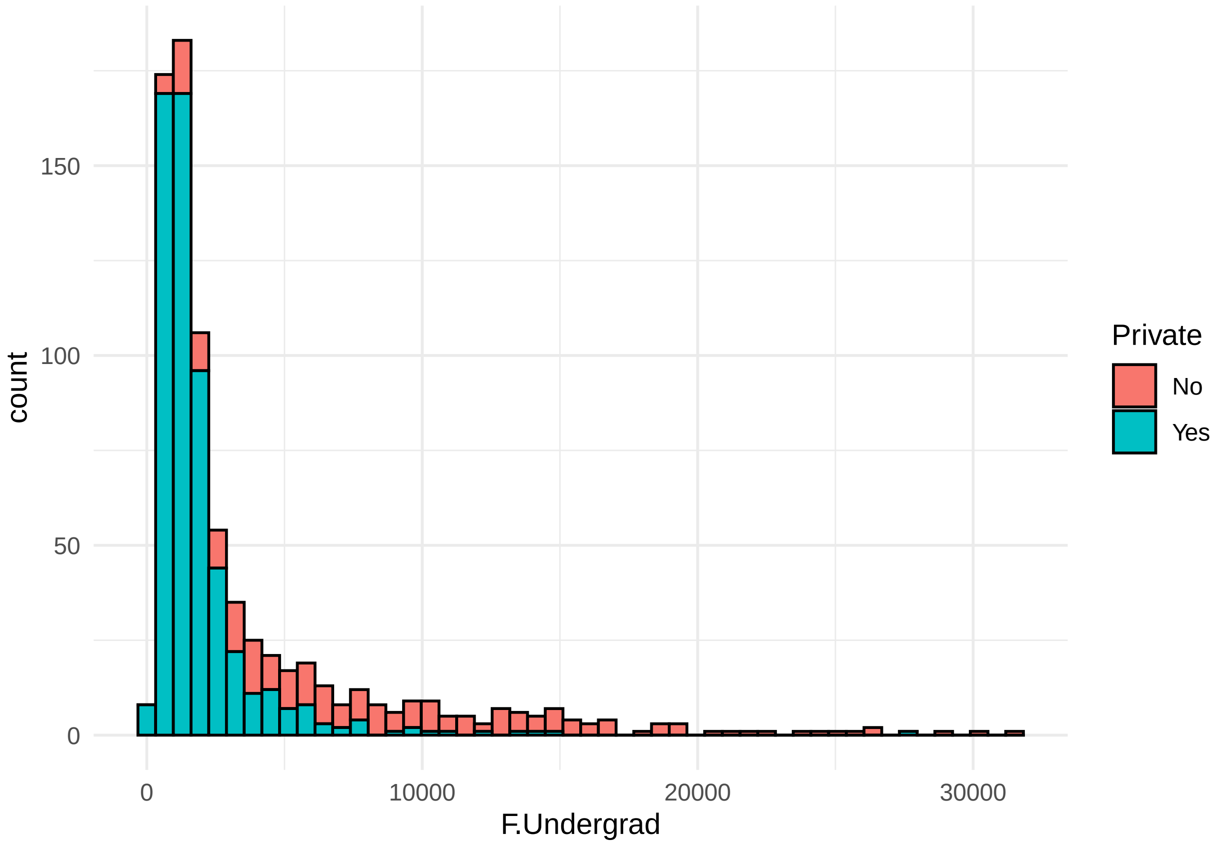
Create a scatterplot of Grad.Rate versus Room.Board, colored by the Private column.

```r
ggplot(df, aes(Room.Board, Grad.Rate)) +
  geom_point(aes(color = Private))
```



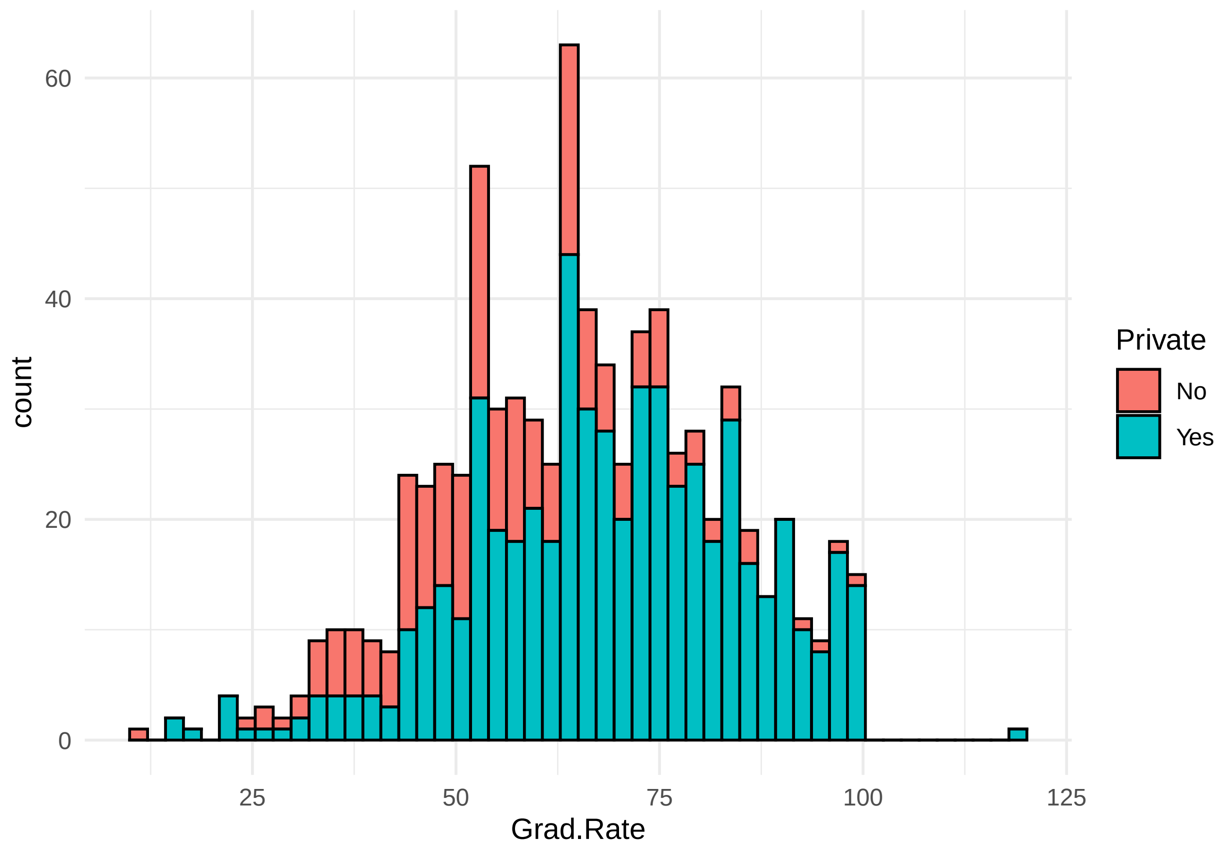
Create a histogram of full time undergrad students, color by Private.

```
ggplot(df, aes(F.Undergrad, fill = Private)) +
  geom_histogram(color = "black", bins = 50)
```



Create a histogram of Grad.Rate colored by Private. You should see something odd here.

```
ggplot(df, aes(Grad.Rate, fill = Private)) +
  geom_histogram(color = "black", bins = 50)
```



What college had a Graduation Rate of above 100% ?

```
rownames(df[df$Grad.Rate > 100, ])
```

```
## [1] "Cazenovia College"
```

Change that college's grad rate to 100%

```
for (i in 1:100){
  if (df$Grad.Rate[i] > 100){
    df$Grad.Rate[i] = 100
  }
}
```

Train Test Split

Split your data into training and testing sets 70/30. Use the `caTools` library to do this.

```
library(caTools)
```

```
## Warning: package 'caTools' was built under R version 3.6.3
```

```r
sample <- sample.split(df$Private, 0.7)

training <- subset(df, sample == T)
testing <- subset(df, sample == F)
```

Decision Tree

Use the rpart library to build a decision tree to predict whether or not a school is Private. Remember to only build your tree off the training data.

```r
library(rpart)

treeModel <- rpart(Private ~ ., training, method = "class")
```

Use `predict()` to predict the Private label on the test data.

```r
predict <- predict(treeModel, testing)
```

Check the Head of the predicted values. You should notice that you actually have two columns with the probabilities.

```r
head(predict)
```

```
##                                            No        Yes
## Alverno College                    0.02040816 0.97959184
## Andrews University                 0.02040816 0.97959184
## Angelo State University            0.94736842 0.05263158
## Antioch University                 0.02040816 0.97959184
## Aquinas College                    0.02040816 0.97959184
## Arizona State University Main campus 0.94736842 0.05263158
```

Turn these two columns into one column to match the original Yes/No Label for a Private column.

```r
predict <- data.frame(predict) %>%
  transmute(Label = ifelse(Yes > No, "Yes", "No"))
```

Now use `table()` to create a confusion matrix of your tree model.

```r
table(as.matrix(predict), testing$Private)
```

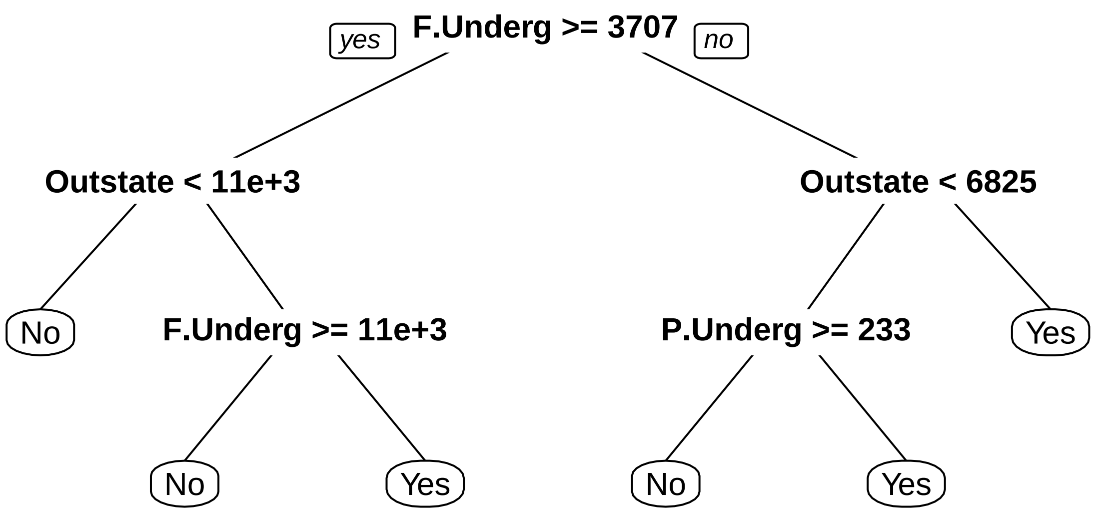
```
##
##          No Yes
##    No    51   8
##    Yes   13 161
```

Use the `rpart.plot` library and the `prp()` function to plot out your tree model.

```r
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 3.6.3
```

```r
prp(treeModel)
```



Random Forest

Now let's build out a random forest model!

Call the `randomForest` package library

```r
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.6.3
```

Now use `randomForest()` to build out a model to predict Private class. Add `importance=TRUE` as a parameter in the model. (Use help(randomForest) to find out what this does.

```r
randomForestModel <- randomForest(Private ~ ., training, importance = T)
```

What was your model's confusion matrix on its own training set? Use `model$confusion`.

```r
randomForestModel$confusion
```

```
##      No Yes class.error
## No  125  23  0.15540541
## Yes  14 382  0.03535354
```

Grab the feature importance with `model$importance`. Refer to the reading for more info on what Gini[1] means.[2]

```r
randomForestModel$importance
```

```
##                      No          Yes MeanDecreaseAccuracy MeanDecreaseGini
## Apps        0.0294866878 0.0134061780         0.0177890305         9.117090
## Accept      0.0370123556 0.0137630721         0.0200677431        14.071934
## Enroll      0.0603772842 0.0317175050         0.0397756054        22.784131
## Top10perc   0.0074196989 0.0037018265         0.0047190439         3.999770
## Top25perc   0.0058087195 0.0021203510         0.0031284366         3.493445
## F.Undergrad 0.1534339586 0.0606102382         0.0858250636        42.745743
## P.Undergrad 0.0391775252 0.0065622905         0.0153484967        16.277205
## Outstate    0.1250765327 0.0508757249         0.0710933695        37.411397
## Room.Board  0.0118341909 0.0168489934         0.0154349221        11.708929
## Books       0.0001775135 0.0001670596         0.0001739772         2.112954
## Personal    0.0047247127 0.0002846135         0.0014912475         3.260299
## PhD         0.0135492166 0.0049833572         0.0072996940         4.692713
## Terminal    0.0063575278 0.0054643805         0.0057177032         4.636212
## S.F.Ratio   0.0347543961 0.0099558720         0.0167312471        18.679402
## perc.alumni 0.0275127440 0.0040052434         0.0103249880         5.889651
## Expend      0.0162117022 0.0079925301         0.0103153212         8.568742
## Grad.Rate   0.0112286512 0.0039036740         0.0058631821         5.231478
```

Predictions

Now use your random forest model to predict on your test set!

```r
predict.rf <- predict(randomForestModel, testing)

table(predict.rf, testing$Private)
```

```
##
## predict.rf  No Yes
##        No   55   3
##        Yes   9 166
```

It should have performed better than just a single tree, how much better depends on whether you are emasuring recall, precision, or accuracy as the most important measure of the model.