Decision Tree and Random Forest Project

Minghao Liu

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

```
library(ISLR)
df <- College
head(df)
```

```
##                              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
```

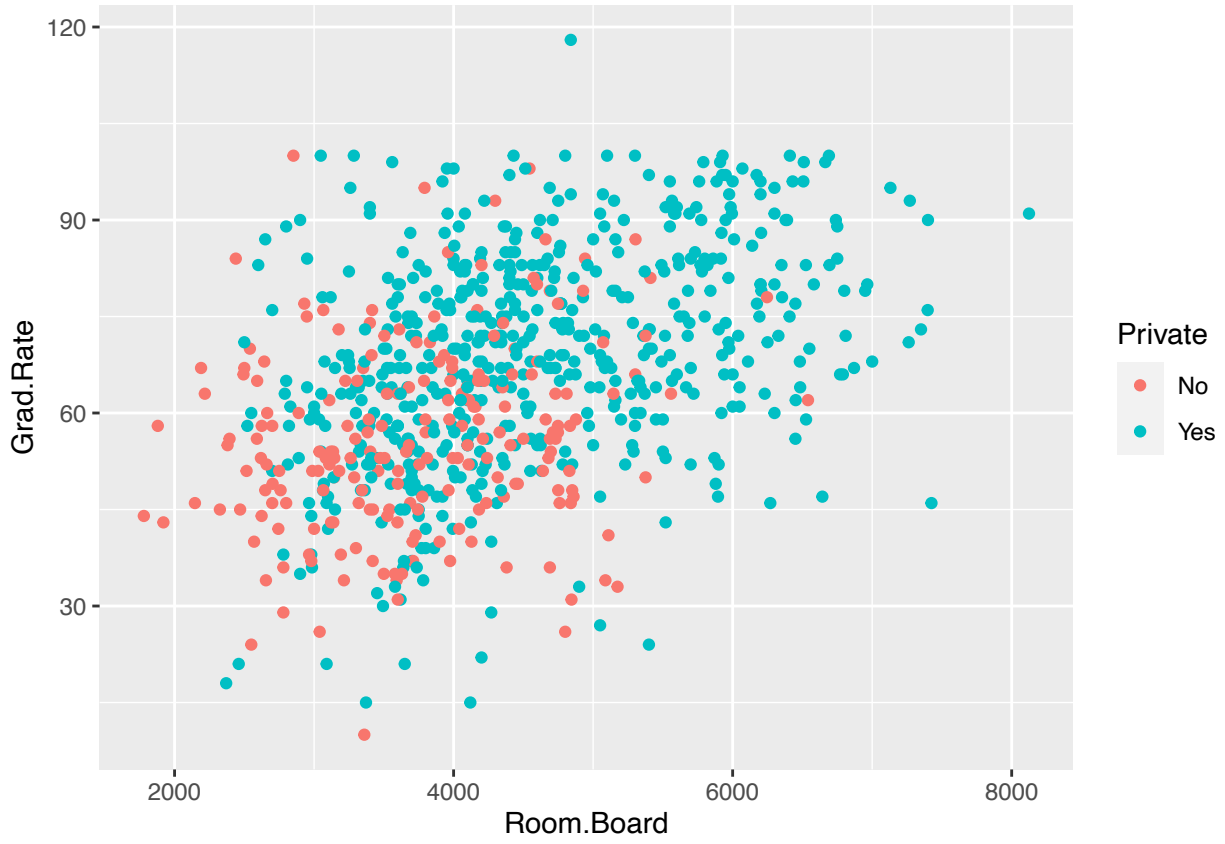
EDA

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

Minghao Liu

```
library(ggplot2)

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

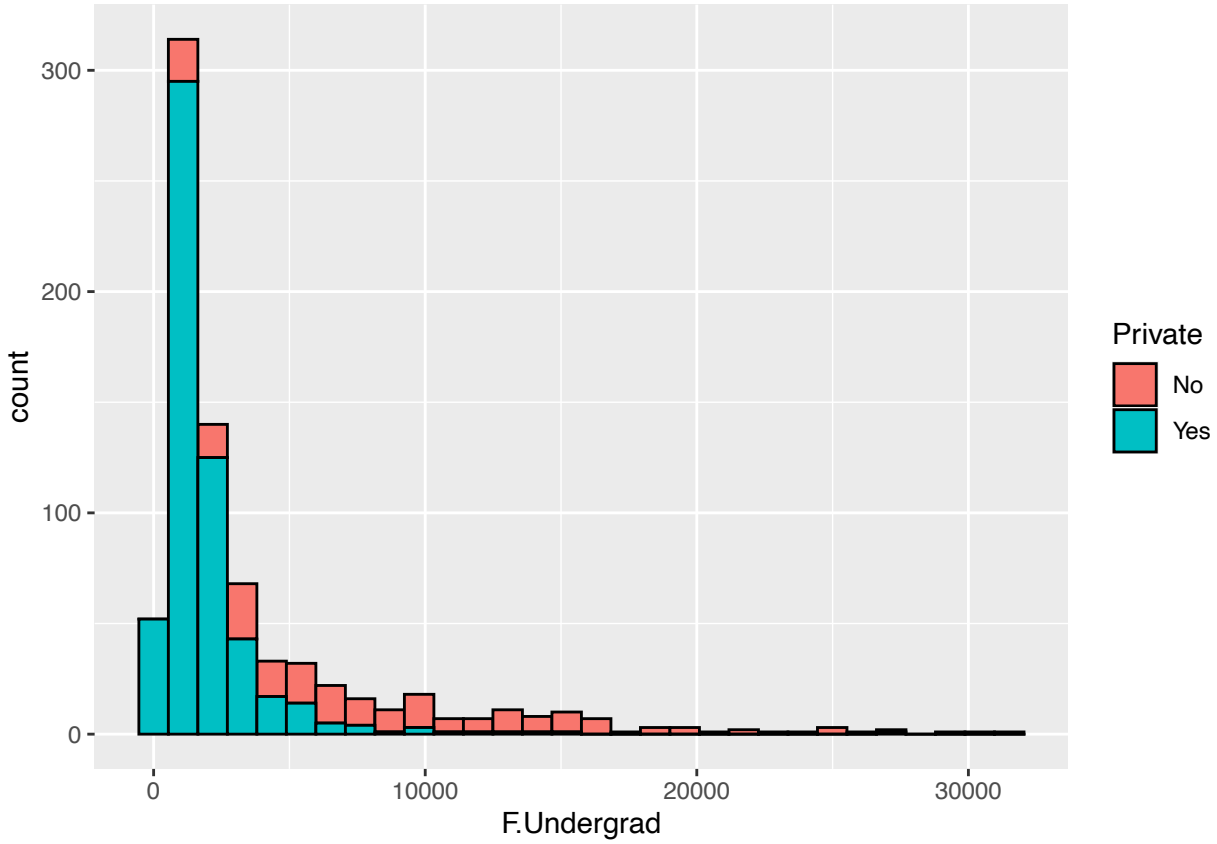


Create a histogram of full time undergrad students, color by Private.

```
ggplot(data = df, aes(x = F.Undergrad)) + geom_histogram(color = 'black', aes(fill = Private))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

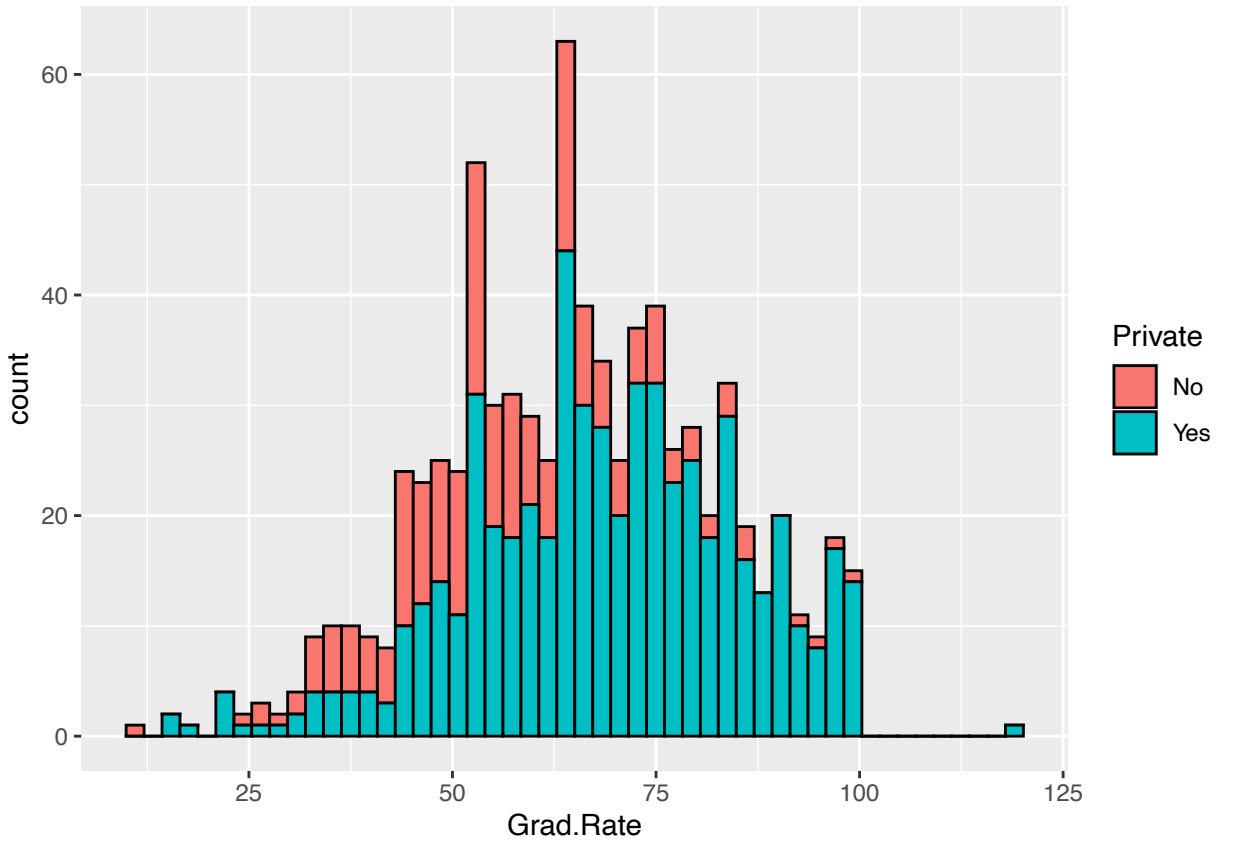

Minghao Liu



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

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


Minghao Liu



What college had a Graduation Rate of above 100% ? Change that college's grad rate to 100%

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
filter(df,df[,'Grad.Rate'] > 100)
```

```
##                   Private Apps Accept Enroll Top10perc Top25perc F.Undergrad
## Cazenovia College     Yes 3847   3433    527         9        35        1010
##                   P.Undergrad Outstate Room.Board Books Personal PhD Terminal
## Cazenovia College          12     9384       4840   600      500  22       47
##                   S.F.Ratio perc.alumni Expend Grad.Rate
## Cazenovia College      14.3          20   7697       118
```


Minghao Liu

```
subset(df, Grad.Rate > 100)
```

```
##                   Private Apps Accept Enroll Top10perc Top25perc F.Undergrad
## Cazenovia College     Yes 3847   3433    527         9        35        1010
##                   P.Undergrad Outstate Room.Board Books Personal PhD Terminal
## Cazenovia College          12     9384       4840   600      500  22       47
##                   S.F.Ratio perc.alumni Expend Grad.Rate
## Cazenovia College      14.3          20   7697       118
```

```
df[row.names(filter(df,df[,'Grad.Rate'] > 100)),'Grad.Rate'] <- 100
```

Train Test Split

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

```
library(caTools)
set.seed(101)

sample <- sample.split(df, SplitRatio = 0.7)
train <- subset(df, sample == TRUE)
test <- subset(df, sample == FALSE)
```

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.

```
library(rpart)

tree <- rpart(Private ~ ., data = train)
```

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

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

```
tree.preds <- predict(tree, test)


head(tree.preds)
```

```
##                                                   No       Yes
## Adrian College                                0.01212121 0.9878788
## Alfred University                             0.01212121 0.9878788
## Allegheny College                             0.01212121 0.9878788
## Allentown Coll. of St. Francis de Sales       0.01212121 0.9878788
## Alma College                                  0.01212121 0.9878788
## Amherst College                               0.01212121 0.9878788
```


Minghao Liu

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

```
typeof(tree.preds)
```

```
## [1] "double"
```

```
tree.preds <- as.data.frame(tree.preds)

classifer <- function(x){
  if (x >= 0.5){
    return('Yes')
  }
  else{
    return('No')
  }
}

tree.preds$Private <- sapply(tree.preds$Yes, classifer)

head(tree.preds)
```

```
##                                                 No       Yes Private
## Adrian College                           0.01212121 0.9878788     Yes
## Alfred University                        0.01212121 0.9878788     Yes
## Allegheny College                        0.01212121 0.9878788     Yes
## Allentown Coll. of St. Francis de Sales 0.01212121 0.9878788     Yes
## Alma College                             0.01212121 0.9878788     Yes
## Amherst College                          0.01212121 0.9878788     Yes
```

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

```
table(test$Private, tree.preds$Private)
```

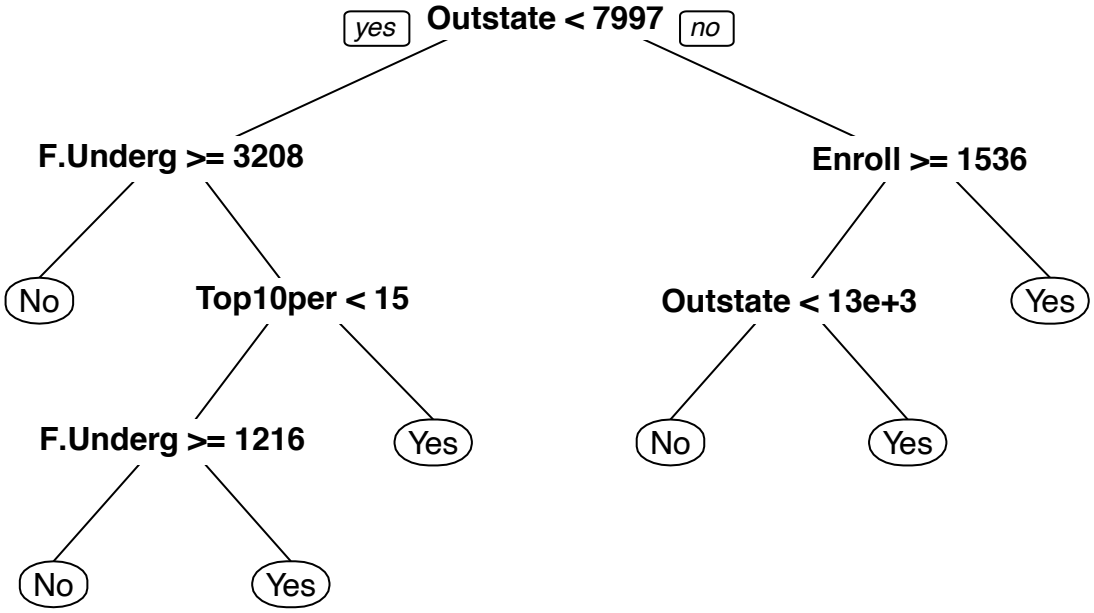
```
##
##        No Yes
##   No   53  14
##   Yes   6 186
```

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

```
library(rpart.plot)

prp(tree)
```


Minghao Liu



Random Forest

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.

```
library(randomForest)
```

```
## randomForest 4.6-14

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

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```


Minghao Liu

```
rf.model <- randomForest(Private ~ ., data = train, importance = T)
```

What was your model's confusion matrix on its own training set?

```
rf.model$confusion
```

```
##      No Yes class.error
## No  133  12  0.08275862
## Yes   9 364  0.02412869
```

Grab the feature importance.

```
rf.model$importance
```

```
##                       No           Yes MeanDecreaseAccuracy MeanDecreaseGini
## Apps         0.024658475  1.176865e-02         0.0154141308         7.625180
## Accept       0.021063137  1.397548e-02         0.0160740943        10.153772
## Enroll       0.054393141  3.498010e-02         0.0403177461        22.235333
## Top10perc    0.012869987  2.113917e-03         0.0051437699         4.577233
## Top25perc    0.011985440  1.872881e-03         0.0047138004         4.500875
## F.Undergrad 0.133073901  6.048939e-02         0.0807754292        39.611262
## P.Undergrad 0.053330538  1.062057e-02         0.0226870104        14.151087
## Outstate     0.151652080  5.187388e-02         0.0795236629        46.554936
## Room.Board   0.036247805  1.523931e-02         0.0210895296        10.982187
## Books        0.001197314  7.109034e-05         0.0004056415         1.917629
## Personal     0.004092538 -6.835476e-05         0.0011230531         3.372707
## PhD          0.008025217  3.900076e-03         0.0050068432         2.934670
## Terminal     0.004555453  2.650241e-03         0.0032332940         2.955144
## S.F.Ratio    0.037624794  4.893113e-03         0.0140725202        15.107404
## perc.alumni 0.030022301  1.671572e-03         0.0095890662         4.668519
## Expend       0.036537073  1.530970e-02         0.0211803028        12.163700
## Grad.Rate    0.013102042  3.623410e-03         0.0062475493         5.007583
```

MeanDecreaseAccuracy: It measures how much inclusion of this predictor in the model reduces classification error.
MeanDecreaseGini: Gini is defined as "inequity" when used in describing a society's distribution of income, or a measure of "node impurity" in tree-based classification. A low Gini (i.e. higher descrease in Gini) means that a particular predictor variable plays a greater role in partitioning the data into the defined classes.

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

```
p <- predict(rf.model, test)

table(p, test$Private)
```

```
##
## p      No Yes
##   No   54   8
##   Yes  13 184
```


Minghao Liu

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.