Statistics 216 Homework 1

Xiangpeng Li

1. Which statistical learning method is performing better? flexible method or inflexible method.

(a) The number of observations `n` is extremely large, and the number of predictiors `p` is small

Flexible method is better, because given a large set of sample size we can make use of it to train the model

(b) The number of predictors `p` is extremely large, and the number of observations `n` is small

Inflexible method is better, because flexible method may overfit the sample data.

(c) The variance of the error terms, i.e. $\sigma^2 = Var(\epsilon)$, is extremely high

Inflexible method is better, because flexible method will overfit a lot error instead of real values.

(d) The relationship between the predictors and response is highly non-linear, and σ^2 is small

Flexible method is better, since the relationship is non-linear, introducing a inflexible method will cause a higher bias.

(e) The relationship between the predictors and response is highly non-linear, and σ^2 is large

It depends on how relatively non-linear and how relatively large σ^2 is. Flexible method will work better in non-linear relationship but a high σ^2 will introduce too much noise.

2. Explain whether each scenario below is a regression, classification or unsupervised learning problem, and indicate whether we are most interested in inference or prediction. Finally, provide `n` and `p`.

(a) We collect a set of data on the top 500 firms in the US. For each firm we record profit, number of employees, industry and the CEO salary. We are interested in understanding which factors affect CEO salary

Regression, inference, n = 500, p = 3

(b) Our website has collected the ratings of 1000 different restaurants by 10,000 customers. Each customer has rated about 100 restaurants, and we would like to recommend restaurants to customers who have not yet been there.

Classification, prediction, n = 10,000 * 100 = 1,000,000, p = 1

(c) We are considering launching a new product and wish to know whether it will be a success or a failure. We collect data on 20 similar products that were previously launched. For each product we have recorded whether it was a success or failure, price charged for the product, marketing budget, competition price, and ten other variables.

Classification, prediction, n = 20, p = 13

(d) We are interesting in predicting the % change in the US dollar in relation to the weekly changes in the world stock markets. Hence we collect weekly data for all of 2012. For each week we record the % change in the dollar, the % change in the US market, the % change in the British market, and the % change in the German market.

Regression, prediction, n = 52, p = 3

3. In this next question we consider some real-life applications of statistical learning

(a)

1). A shopping mall wants to predict whether male or female is going to spend more money during shopping. They record last 5 years sales, shopping frequency, time. All those data are per gender.
Response: male or female
Predictors: sales, shopping frequency, time
Goal: Prediction
2). A rating agency will rate a stock between AAA to DDD. In order to do that they record the company sales, number of employees, previous ratings in 5 years.
Response: ratings
Predictors: company sales, number of employees, previous ratings
Goal: prediction
3). Whether my application to Stanford University will be approved or rejected.
Response: Approve or reject
Predictors: GPA, working experience, research experience
Goal: prediction

(b)

1). A fast-food restaurant wants to predict how much revenue they can make in the next year. They collect last year weekly records. For each week it has advertising cost, personnel cost, material cost and revenue.
Response: next year revenue
Predictors: advertising cost, personnel cost, material cost
Goal: prediction
2). Youtube wants to know which factors impact on the time people spending on a video. They have 10000 video sample. For each video they collect the category of that video, length of video, whether they have inserted ads in between, number of subscriber of that youbuter.
Response: time spent on a video
Predictors: the category of that video, length of video, whether they have inserted ads, number of subscriber of a youbuter
Goal: Inference
3). Birth rate in U.S
Response: Birth Rate
Predictors: number of hospitals, number of people who are married, house income
Goal: prediction

(c)

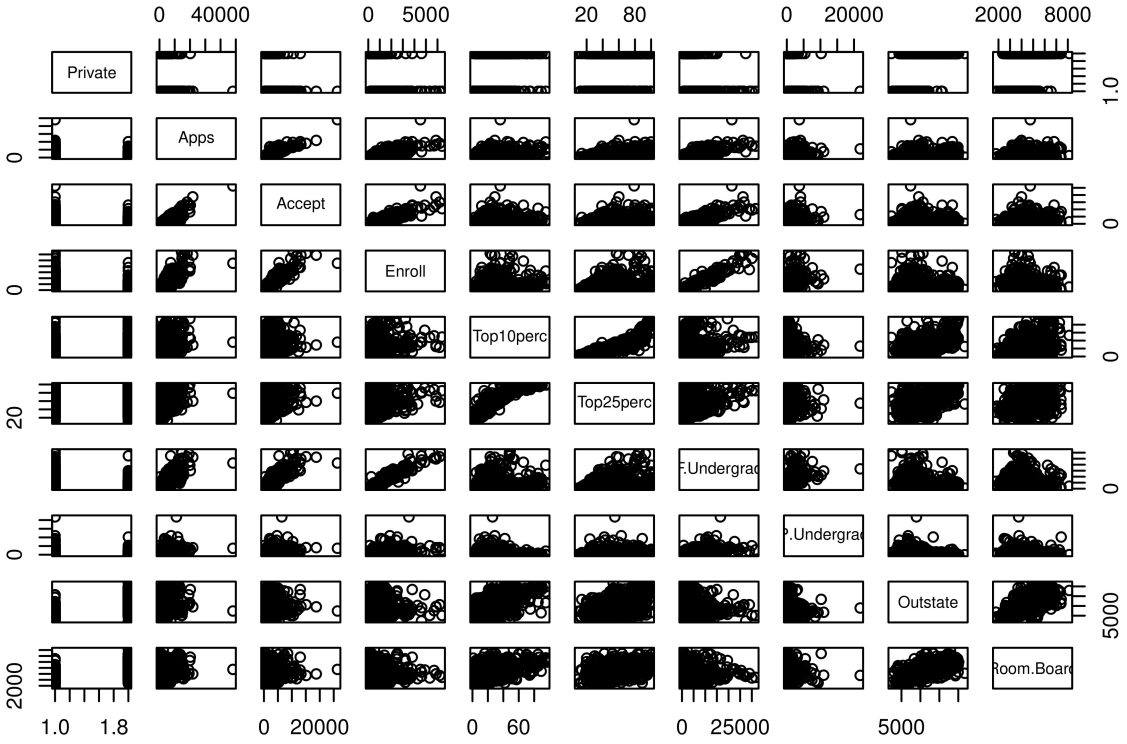
1). Banks want to find divide their credit card holders into different groups based on their spending behaviors such as monthly balance, FICO scores, income.
2). A restaurant wants to divide their customer into different groups based on their food preference, time spent during restaurant, gender.
3). A univertisy wants to cluster their students into different group based on their GPA, major, research experience."

4. This exercise relates to the College data set, which can be found in the file College.csv. It contains a number of variables for 777 different universities and colleges in the US.

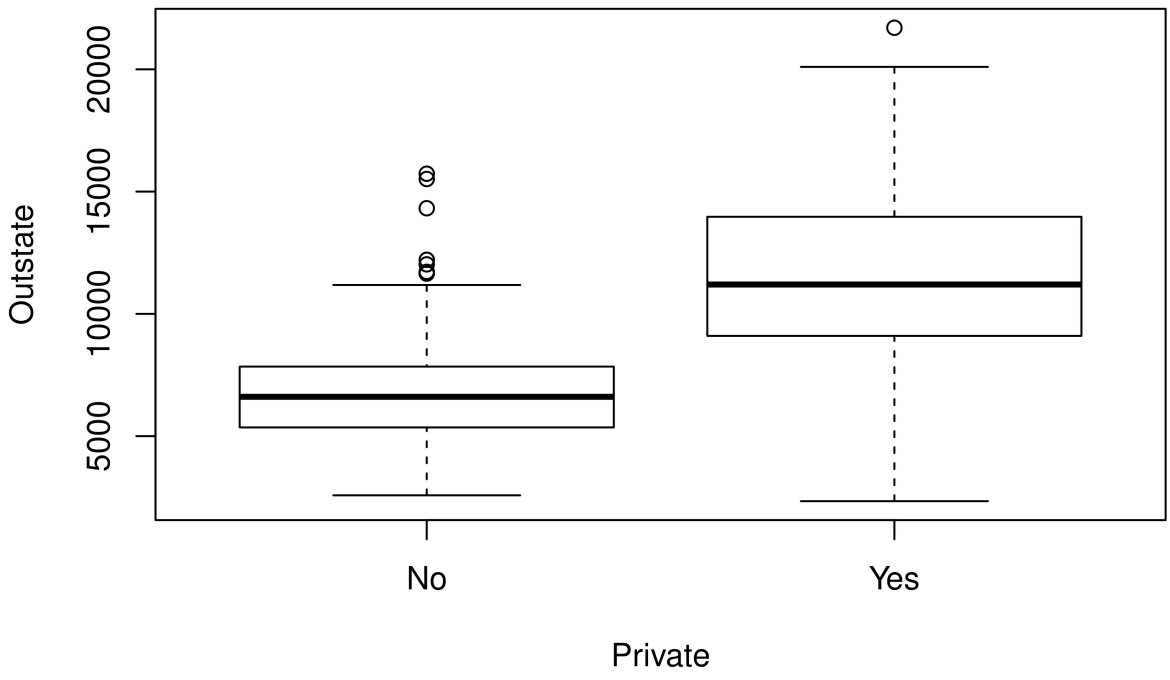
```
setwd("D:/One Drive/OneDrive/Document/Study/Stanford/Introduction to Statistical Learning/homework/hw1")
college = read.csv("college.csv")
rownames(college)=college[,1]
college=college[,-1]
summary(college)
```

```
##  Private        Apps           Accept          Enroll       Top10perc
##  No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00
##  Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00
##            Median : 1558   Median : 1110   Median : 434   Median :23.00
##            Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56
##            3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00
##            Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00
##    Top25perc      F.Undergrad     P.Undergrad         Outstate
##  Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340
##  1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320
##  Median : 54.0   Median : 1707   Median :  353.0   Median : 9990
##  Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441
##  3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925
##  Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700
##    Room.Board       Books           Personal         PhD
##  Min.   :1780   Min.   :  96.0   Min.   : 250   Min.   :  8.00
##  1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00
##  Median :4200   Median : 500.0   Median :1200   Median : 75.00
##  Mean   :4358   Mean   : 549.4   Mean   :1341   Mean   : 72.66
##  3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00
##  Max.   :8124   Max.   :2340.0   Max.   :6800   Max.   :103.00
##     Terminal       S.F.Ratio      perc.alumni        Expend
##  Min.   : 24.0   Min.   : 2.50   Min.   : 0.00   Min.   : 3186
##  1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00   1st Qu.: 6751
##  Median : 82.0   Median :13.60   Median :21.00   Median : 8377
##  Mean   : 79.7   Mean   :14.09   Mean   :22.74   Mean   : 9660
##  3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10830
##  Max.   :100.0   Max.   :39.80   Max.   :64.00   Max.   :56233
##    Grad.Rate
##  Min.   : 10.00
##  1st Qu.: 53.00
##  Median : 65.00
##  Mean   : 65.46
##  3rd Qu.: 78.00
##  Max.   :118.00
```

```r
pairs(college[, 1:10])
```



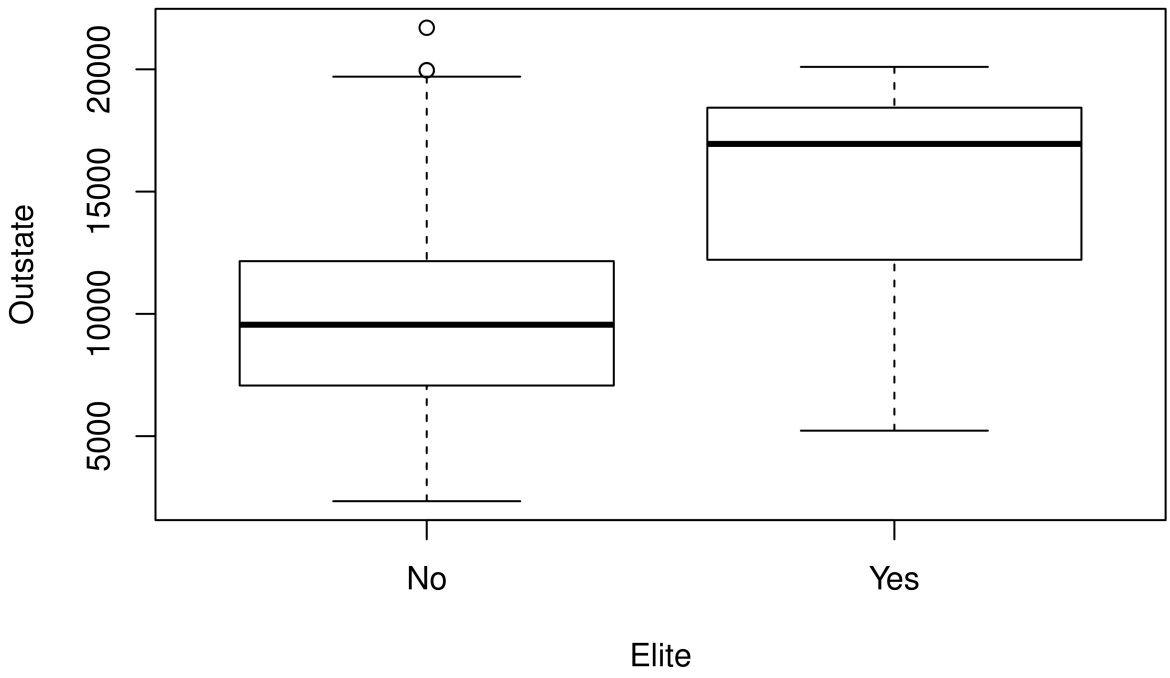

```r
plot(college$Private, college$Outstate, xlab = "Private", ylab = "Outstate")
```



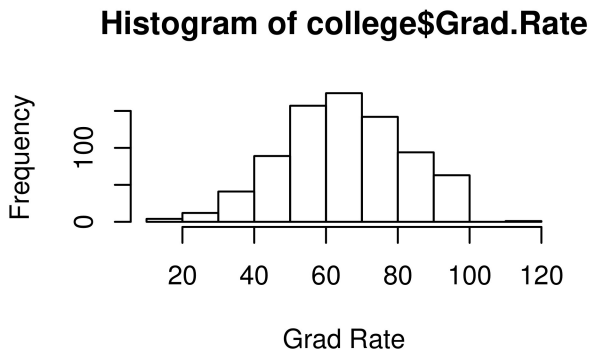

```
Elite = rep("No",nrow(college))
Elite[college$Top10perc >50] = "Yes"
Elite = as.factor(Elite)
college = data.frame(college ,Elite)
summary(college$Elite)
```

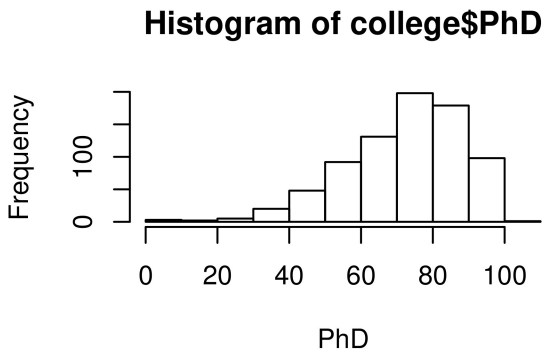
```
##  No Yes
## 699  78
```

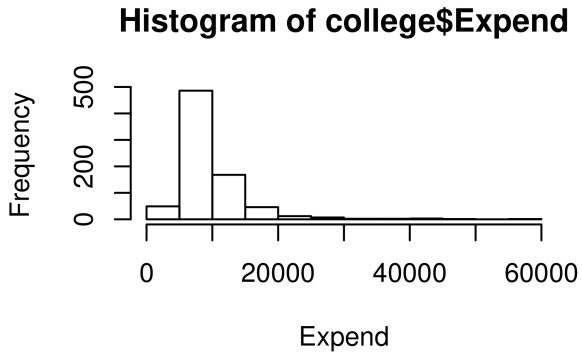
```
plot(college$Elite, college$Outstate, xlab = "Elite", ylab = "Outstate")
```

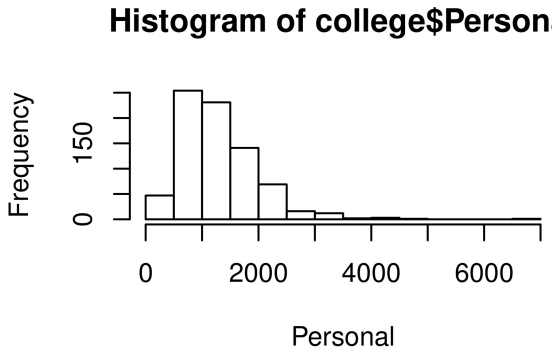



```
par(mfcol = c(2, 2))
hist(college$Grad.Rate, xlab = "Grad Rate", ylab = "Frequency")
hist(college$Expend, xlab = "Expend", ylab = "Frequency")
hist(college$PhD, xlab = "PhD", ylab = "Frequency")
hist(college$Personal, xlab = "Personal", ylab = "Frequency")
```









5. In this exercise, we will predict the number of applications received using the other variables in the College data set

(a)

```
indices <- split(sample(nrow(college), nrow(college), replace=FALSE), as.factor(1:2))
trainingSet = college[indices[[1]], ]
testSet = college[-indices[[1]], ]
```

(b)

```
fit <- lm(Apps ~ . - Accept - Enroll - Elite, data = trainingSet)
summary(fit)
```

```
##
## Call:
## lm(formula = Apps ~ . - Accept - Enroll - Elite, data = trainingSet)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -9950   -828   -126    580  32239
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.440e+03  1.257e+03  -2.736 0.006513 **
```

```
## PrivateYes  -4.636e+02  4.130e+02  -1.123 0.262321
## Top10perc    9.363e+00  1.767e+01   0.530 0.596461
## Top25perc    5.752e+00  1.425e+01   0.404 0.686667
## F.Undergrad  6.740e-01  3.625e-02  18.593  < 2e-16 ***
## P.Undergrad -9.470e-02  8.669e-02  -1.092 0.275353
## Outstate     1.962e-02  5.616e-02   0.349 0.727016
## Room.Board   3.098e-01  1.505e-01   2.059 0.040195 *
## Books        8.332e-02  6.717e-01   0.124 0.901345
## Personal    -8.791e-02  2.077e-01  -0.423 0.672371
## PhD         -9.144e+00  1.402e+01  -0.652 0.514599
## Terminal    -3.497e+00  1.533e+01  -0.228 0.819679
## S.F.Ratio    3.051e+01  4.087e+01   0.747 0.455770
## perc.alumni -2.767e+01  1.272e+01  -2.175 0.030226 *
## Expend       1.641e-01  4.318e-02   3.801 0.000168 ***
## Grad.Rate    2.986e+01  9.083e+00   3.288 0.001105 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2312 on 373 degrees of freedom
## Multiple R-squared:  0.7149, Adjusted R-squared:  0.7034
## F-statistic: 62.36 on 15 and 373 DF,  p-value: < 2.2e-16
```

We are using `training MSE` and `test MSE` to measure the quality of fit.

```
trainingMSE = mean(fit$residuals^2)
testMSE = mean((testSet$Apps - predict.lm(fit, testSet)) ^ 2)
summary(trainingMSE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 5124859 5124859 5124859 5124859 5124859 5124859
```

```
summary(testMSE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 2332046 2332046 2332046 2332046 2332046 2332046
```

(c)

As we can see above `testMSE` and `trainingMSE` have large numbers. Also R^2 is 0.8105 using this linear model. We can conclude that linear model does not fit this data very well. However F-statistic is 106.4 which is far more than 1, it suggests at least one of the factors must be related to `Apps`.
`F.Undergrad`, `Room.Board`, `Grad.Rate` and `Private - Yes` have the smallest p-values and are the most important factors. `Top10prec`, `perc.alumni` and `Expend` are the tier 2 important factors.

6. Using the same setup as in the previous question, form a new outcome variable Y which equals one if the number of applications is greater than or equal to the overall median and zero otherwise. Fit a logistic regression model to Y and report the training and test misclassification rates, and the most important predictors. As above, do not include the Elite predictor, or the Accept or Enrol predictors in the regression. Compare the results of this analysis to that of the linear regression approach in the previous question.

```
med = median(college$Apps)
Y = rep(0, nrow(college))
Y[college$Apps >= med] = 1
Y = as.factor(Y)
```

```r
college = data.frame(college, Y)

## exlcude unwanted factors
college = subset(college, select = -c(Accept, Enroll, Elite, Apps))

indices <- split(sample(nrow(college), nrow(college), replace=FALSE), as.factor(1:2))
trainingSet = college[indices[[1]], ]
testSet = college[-indices[[1]], ]
fit <- glm(formula = Y ~ ., family = binomial, data = trainingSet)
summary(fit)
```

```
##
## Call:
## glm(formula = Y ~ ., family = binomial, data = trainingSet)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.95148  -0.27285   0.00000   0.05429   2.91591
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.031e+01  2.964e+00  -3.480 0.000502 ***
## PrivateYes  -3.145e-02  9.307e-01  -0.034 0.973044
## Top10perc    1.342e-02  3.415e-02   0.393 0.694310
## Top25perc   -2.432e-03  2.940e-02  -0.083 0.934084
## F.Undergrad  2.970e-03  4.377e-04   6.785 1.16e-11 ***
## P.Undergrad  1.750e-04  5.422e-04   0.323 0.746869
## Outstate     1.843e-04  1.000e-04   1.842 0.065468 .
## Room.Board   4.506e-04  2.804e-04   1.607 0.108057
## Books       -9.652e-04  1.355e-03  -0.712 0.476281
## Personal     4.792e-04  4.459e-04   1.075 0.282447
## PhD          2.554e-02  2.489e-02   1.026 0.304888
## Terminal    -3.687e-03  2.738e-02  -0.135 0.892881
## S.F.Ratio   -1.204e-01  1.174e-01  -1.026 0.305032
## perc.alumni  7.237e-03  2.465e-02   0.294 0.769038
## Expend      -3.996e-05  1.056e-04  -0.378 0.705117
## Grad.Rate    9.751e-03  1.819e-02   0.536 0.591976
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 538.69  on 388  degrees of freedom
## Residual deviance: 154.35  on 373  degrees of freedom
## AIC: 186.35
##
## Number of Fisher Scoring iterations: 9
```

```r
## calculate training misclassification rate
trainingProbs = predict(fit, type = "response")
trainingPred = rep(0, nrow(trainingSet))
trainingPred[trainingProbs > 0.5] = 1
table(trainingPred, trainingSet$Y)
```

```
##
## trainingPred   0   1
##            0 176  14
##            1  11 188
```

```
## training misclassification rate
1 - mean(trainingPred == trainingSet$Y)
```

```
## [1] 0.06426735
```

```
## calculate test misclassification rate
testProbs = predict(fit, newdata = testSet, type = "response")
testPred = rep(0, nrow(testSet))
testPred[testProbs > 0.5] = 1
table(testPred, testSet$Y)
```

```
##
## testPred   0   1
##        0 184  16
##        1  17 171
```

```
## test misclassification rate
1 - mean(testPred == testSet$Y)
```

```
## [1] 0.08505155
```

The error rates for training set and test set are both at range 6% - 9% which fits better than the linear model. The most important factors are `F.Undergrad`, `Outstate` and `Grad.Rate`. Compared to linear model they both have `F.undergrad` and `Grad.Rate` so we can conclude these two factors are most important to the `Apps`