IEOR HW#3

① a) $C_{imp}(T_{old}) - C_{imp}(T_{new})$

$$= \sum_{n-1}^{m} N Q_n(T_{old}) - \sum_{m+1}^{m+1} \hat{N}_m \hat{Q}_m(T_n) = \Delta$$

$$= \sum_{1}^{m} N_m \left(\frac{1}{N_m}\right) \sum (y - \hat{y}_m)^2 - \sum_{m+1}^{m+1} (y_i - \hat{y}_m)^2$$

$$= \sum_{i=R} (y_i - \hat{y}_m)^2 - \sum_{n=m}^{m+1} (y_i - y_m)^2$$

$$\Delta = \sum (y_i - \hat{y}_m)^2 - \sum (y_i - \hat{y}_m)^2 - \sum (y_i - \hat{y}_m)^2$$

$$\boxed{\begin{aligned}\Delta = &\sum_{i=x\in M_{old}} \left(y_i - \frac{1}{N_m}\sum y_i\right)^2 - \sum \left(y_i - \frac{1}{N_m}\sum y_i\right)^2 \\ &- \sum \left(y_i - \frac{1}{N_{m+1}} \sum y_i\right)^2\end{aligned}}$$

b) $\Delta = \left[\sum (y_i - \hat{y}_m)^2\right] - \left[\sum (y_i - \hat{y}_m)^2\right] - \left[\sum (y_i - \hat{y}_{m-1})^2\right]$

$$= \left[\sum_{i=m_i\in m} (y_i - \hat{y}_m)^2 + \sum (y_i - \hat{y}_m)^2 - \left[\sum (y_i - \hat{y}_m)^2\right]\right.$$

$$- \left[\sum (y_i - y_{m+1})^2\right]$$

$$\underbrace{\sum (y_i - \hat{y}_m)^2 - \left[\sum (y_i - \hat{y}_m)^2\right]}_{\min \hat{R}_m = \hat{y}_m \geq 0} + \underbrace{\sum (y_i - \hat{y}_m)^2 - \left[\sum (y_i - \hat{y}_{m+1})^2\right]}_{\min \hat{R}_{m+1} = \hat{Y}_{m+1} \geq 0}$$

ⓒ $C_\alpha(T_{new}) = C_{imp}(T_{new}) + \alpha SST \cdot |T_{new}|$

$C_\alpha(T_{old}) = C_{imp}(T_{old}) + \alpha SST \cdot |T_{old}|$

$= C_{imp}(T_{old}) + \alpha \sum_{i=1}^{n} (y_i - \hat{y})^2 \cdot |T_{old}|$

$= C_{imp}(T_{old}) + \alpha \sum (y_i - \hat{y})^2 - M$

$C_\alpha(T_{old})\ C_\alpha(T_{new}) = C_{imp}(T_{old}) - C_{imp}(T_{new}) - \alpha \left(\sum (y_i - \hat{y})^2\right)$

$= \Delta - \alpha \left(\sum (y_i - \hat{y})^2\right)$

$C_\alpha(T_{old}) - C_\alpha(T_{new}) = \sum (y_i - \hat{y}_{m_{old}})^2 - \sum (y_i - \hat{y}_{m_{new}})^2$

$- \sum (y_i - \hat{y}_{m_{new}})^2 - \alpha \sum (y_i - \bar{y})^2$

$\alpha \leq R^2_{new} - R^2_{old}$

then

$\sum (y_i - \hat{y}_{m_{old}})^2 - \left[\sum (y_i - \hat{y}_{m_{new}})^2 + \sum (y_i - \hat{y}_{m_{new}})^2 \right.$

$$\boxed{C_\alpha(T_{new}) \leq C_\alpha(T_{old})}$$

```r
library(GGally)
```

```
## Loading required package: ggplot2

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
library(ROCR)
```

```
## Loading required package: gplots

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess
```

```r
library(car)
```

```
## Loading required package: carData
```

```r
library(dplyr)
```

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

## The following object is masked from 'package:car':
##
##     recode

## The following object is masked from 'package:GGally':
##
##     nasa

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

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

```r
library(ggplot2)
library(caTools)
library(rpart)
library(rpart.plot)
library(caret)
```

```
## Loading required package: lattice
```

```r
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
```

```r
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
library(gbm)
```

```
## Loaded gbm 2.1.5
```

2)

a)

```r
set.seed(456)

library(readr)
Letters <- read_csv("C:/Users/Murtz.Kizilbash/Desktop/ieor142/hw3/Letters.csv")
```

```
## Parsed with column specification:
## cols(
##   letter = col_character(),
##   xbox = col_double(),
##   ybox = col_double(),
##   width = col_double(),
##   height = col_double(),
##   onpix = col_double(),
##   xbar = col_double(),
##   ybar = col_double(),
##   x2bar = col_double(),
##   y2bar = col_double(),
##   xybar = col_double(),
##   x2ybar = col_double(),
##   xy2bar = col_double(),
##   xedge = col_double(),
##   xedgeycor = col_double(),
##   yedge = col_double(),
##   yedgexcor = col_double()
## )
```

```r
head(letters)
```

```
## [1] "a" "b" "c" "d" "e" "f"
```

```r
Letters$isB <- as.factor(Letters$letter == "B")
train.ids = sample(nrow(Letters), 0.65*nrow(Letters))
Letters.train = Letters[train.ids,]
Letters.test = Letters[-train.ids,]

table(Letters.train$isB)
```

```
##
## FALSE  TRUE
##  1562   463
```

```r
table(Letters.test$isB)
```

```
##
## FALSE  TRUE
##   788   303
```

i)

```r
Letters$isB = factor(Letters$letter=="B")
spl = sample.split(Letters$isB, SplitRatio = 0.5)
train = subset(Letters, spl)
test = subset(Letters, !spl)
"the accuracy of the baseline method is:"
```

```
## [1] "the accuracy of the baseline method is:"
```

```r
1 - mean(test$isB == "TRUE")
```

```
## [1] 0.754172
```

ii)

```r
mod <- glm(isB ~ xbox + ybox + width + height + onpix + xbar + ybar + x2bar + y2bar + xybar + x2ybar + 

summary(mod)
```

```
##
## Call:
## glm(formula = isB ~ xbox + ybox + width + height + onpix + xbar +
##     ybar + x2bar + y2bar + xybar + x2ybar + xy2bar + xedge +
##     xedgeycor + yedge + yedgexcor, family = "binomial", data = Letters.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1461  -0.1667  -0.0212  -0.0003   3.5412
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -14.771862   2.518821  -5.865 4.50e-09 ***
## xbox         -0.008722   0.119921  -0.073 0.942018
## ybox          0.063592   0.085702   0.742 0.458081
## width        -1.130490   0.150691  -7.502 6.28e-14 ***
## height       -0.795831   0.138778  -5.735 9.78e-09 ***
## onpix         0.889499   0.130406   6.821 9.04e-12 ***
## xbar          0.546162   0.132610   4.119 3.81e-05 ***
## ybar         -0.573137   0.113890  -5.032 4.84e-07 ***
## x2bar        -0.334427   0.097979  -3.413 0.000642 ***
## y2bar         1.416933   0.132082  10.728  < 2e-16 ***
## xybar         0.290159   0.088709   3.271 0.001072 **
## x2ybar        0.553462   0.124573   4.443 8.88e-06 ***
## xy2bar       -0.377165   0.104077  -3.624 0.000290 ***
## xedge        -0.248985   0.094499  -2.635 0.008419 **
## xedgeycor     0.078641   0.101424   0.775 0.438125
## yedge         1.648100   0.125743  13.107  < 2e-16 ***
## yedgexcor     0.303859   0.071467   4.252 2.12e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2177.40  on 2024  degrees of freedom
## Residual deviance:  643.83  on 2008  degrees of freedom
## AIC: 677.83
##
## Number of Fisher Scoring iterations: 8
```

```r
vif(mod)
```

```
##      xbox      ybox     width    height     onpix      xbar      ybar
##  5.495921  7.937591  7.746360  8.745511  7.833037  2.896704  1.925650
##     x2bar     y2bar     xybar    x2ybar    xy2bar     xedge xedgeycor
##  2.614307  1.876439  2.884082  2.712858  2.207120  3.029484  1.828222
##     yedge yedgexcor
##  4.153572  1.690583
```

```
predtest = predict(mod, Letters.test, type = 'response')
summary(predtest)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000000 0.0002273 0.0208780 0.2762650 0.6402234 0.9996789
```

iii)

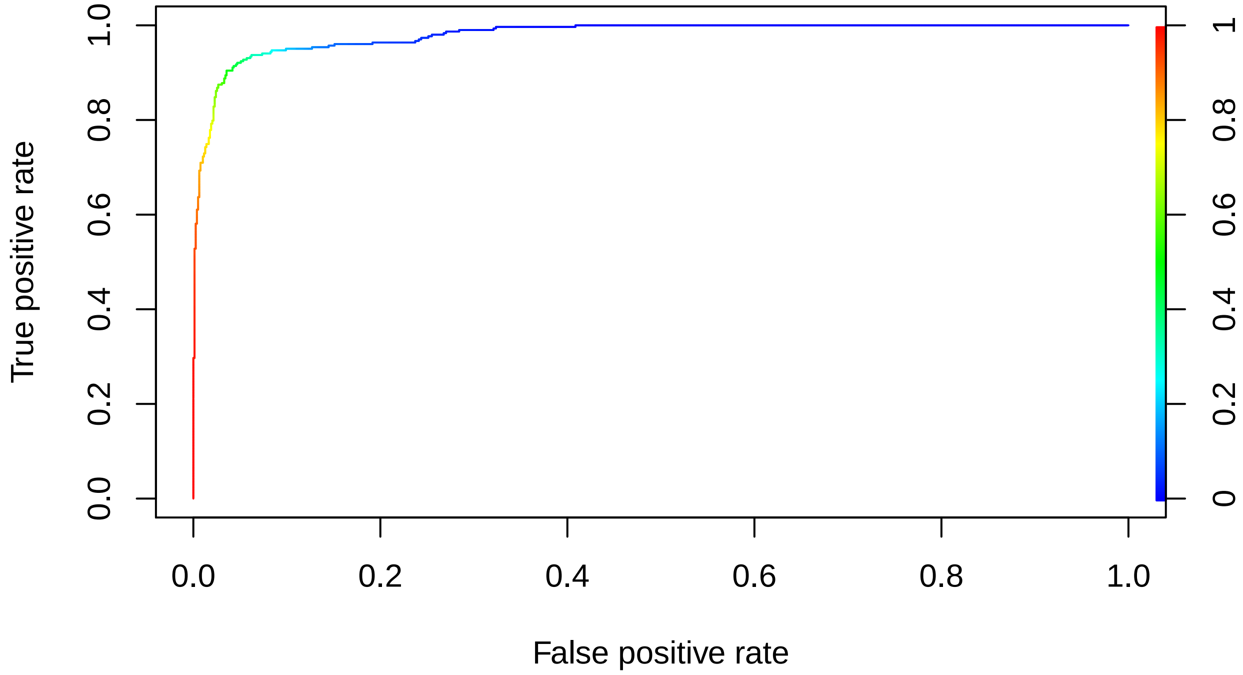
```
table(Letters.test$isB, predtest > 0.5)
```

```
##
##         FALSE TRUE
##   FALSE   760   28
##   TRUE     30  273
```

```
log.pred = prediction(predtest, Letters.test$isB)

logperf = performance(log.pred, 'tpr', 'fpr')

plot(logperf, colorize = TRUE)
```



```r
print('the auc is:')
```

```
## [1] "the auc is:"
```

```r
as.numeric(performance(log.pred, 'auc')@y.values)
```

```
## [1] 0.9796661
```

iv)

```r
CARTb <- rpart(isB ~ . - letter, data = train, method='class')
CARTb_predict <- predict(CARTb, newdata = test, type = "class")
table(test$isB, CARTb_predict)
```

```
##        CARTb_predict
##         FALSE TRUE
##   FALSE  1130   45
##   TRUE     77  306
```

```r
" "
```

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

```r
"the accuracy of the CART model on the test set, is:"
```

```
## [1] "the accuracy of the CART model on the test set, is:"
```

```r
cartModelAccuracy = (1121+329) / nrow(test)
cartModelAccuracy
```

```
## [1] 0.9306804
```

v)

```r
#DONE
#install.packages("randomForest")

m2 = randomForest(isB ~ . - letter, train)
pred <- predict(m2, newdata = test, type = "class")
table(test$isB, pred)
```

```
##        pred
##         FALSE TRUE
##   FALSE  1160   15
##   TRUE     30  353
```

```
" "
```

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

```
"[Part v] The accuracy of the Random Forest Model on the test set is:"
```

```
## [1] "[Part v] The accuracy of the Random Forest Model on the test set is:"
```

```
randomForestAccuracy = (1158+361) / nrow(test)
randomForestAccuracy
```

```
## [1] 0.9749679
```

vi)

```
"CART Model Accuracy = "
```

```
## [1] "CART Model Accuracy = "
```

```
cartModelAccuracy
```

```
## [1] 0.9306804
```

```
""
```

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

```
"Random Forest Model Accuracy = "
```

```
## [1] "Random Forest Model Accuracy = "
```

```
randomForestAccuracy
```

```
## [1] 0.9749679
```

```
"Comparing the accuracy of the logistic regression, CART, and Random Forest Models, the one that perfor
```

```
## [1] "Comparing the accuracy of the logistic regression, CART, and Random Forest Models, the one that
```

b)

(i)

```
spl = sample.split(Letters$isB, SplitRatio = 0.5)
train = subset(Letters, spl)
test = subset(Letters, !spl)

table(test$letter)
```

```
##
##   A   B   P   R
## 399 383 405 371
```

```
"The baseline model predicts P as the most frequent result."
```

```
## [1] "The baseline model predicts P as the most frequent result."
```

```
"The baseline accuracy is = "
```

```
## [1] "The baseline accuracy is = "
```

```
401 / nrow(test)
```

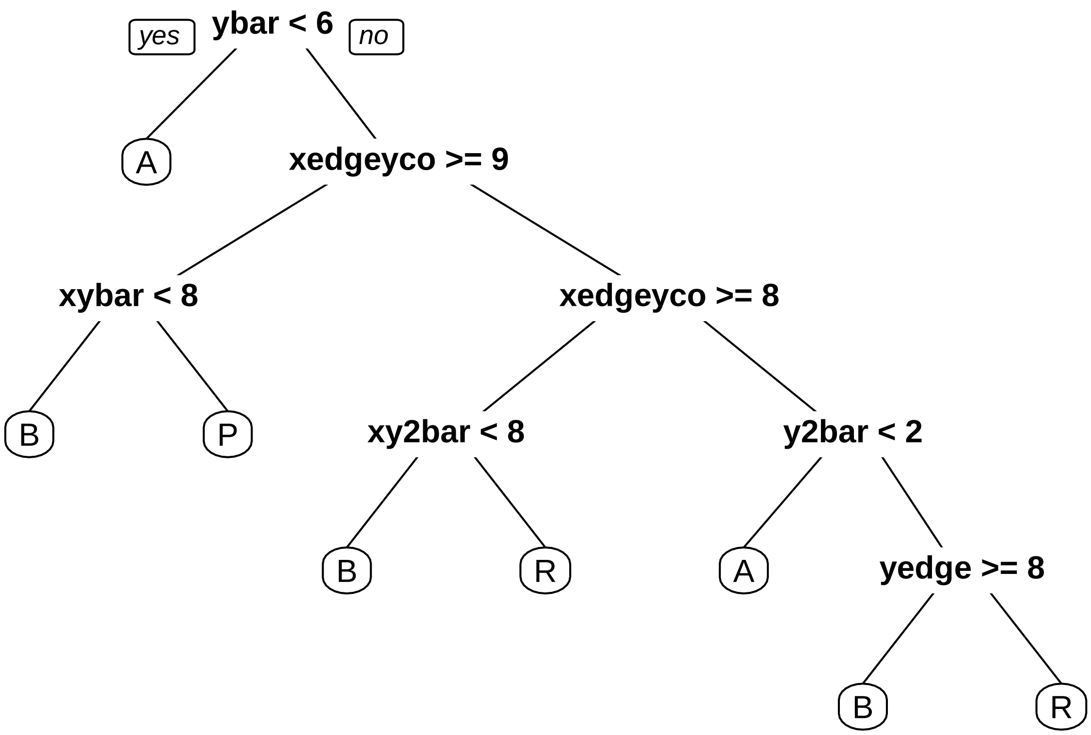
```
## [1] 0.2573813
```

(ii)

```
#LDA Model.
```

(iii)

```
CARTb <- rpart(letter ~ . - isB, data = train, method='class')
prp(CARTb)
```



```
CARTb_predict <- predict(CARTb, newdata = test, type = "class")
length(CARTb_predict)
```

```
## [1] 1558
```

```
table(test$letter, CARTb_predict)
```

```
##    CARTb_predict
##       A   B   P   R
##   A 358  20   0  21
##   B  17 284  18  64
##   P   5  34 362   4
##   R   8  41   8 314
```

```
" "
```

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

```
"The test set accuracy of my CART model is ="
```

```
## [1] "The test set accuracy of my CART model is ="
```

```
(355+237+377+327)/1558
```

```
## [1] 0.8318357
```

(v)