

Lab_4

Register Number:19BLC1186

Name:Tarun Sidhu

Lab Exercise No:4

Date:22/2/2021

Dataset: patients_data

Task: For the given dataset carry out an analysis using Linear Discriminant Analysis and develop a model for multiclass classification. Present the ROC curve and AUC based performance metrics.

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##  
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':  
##  
##      cov, smooth, var
```

```
library(randomForest)
```

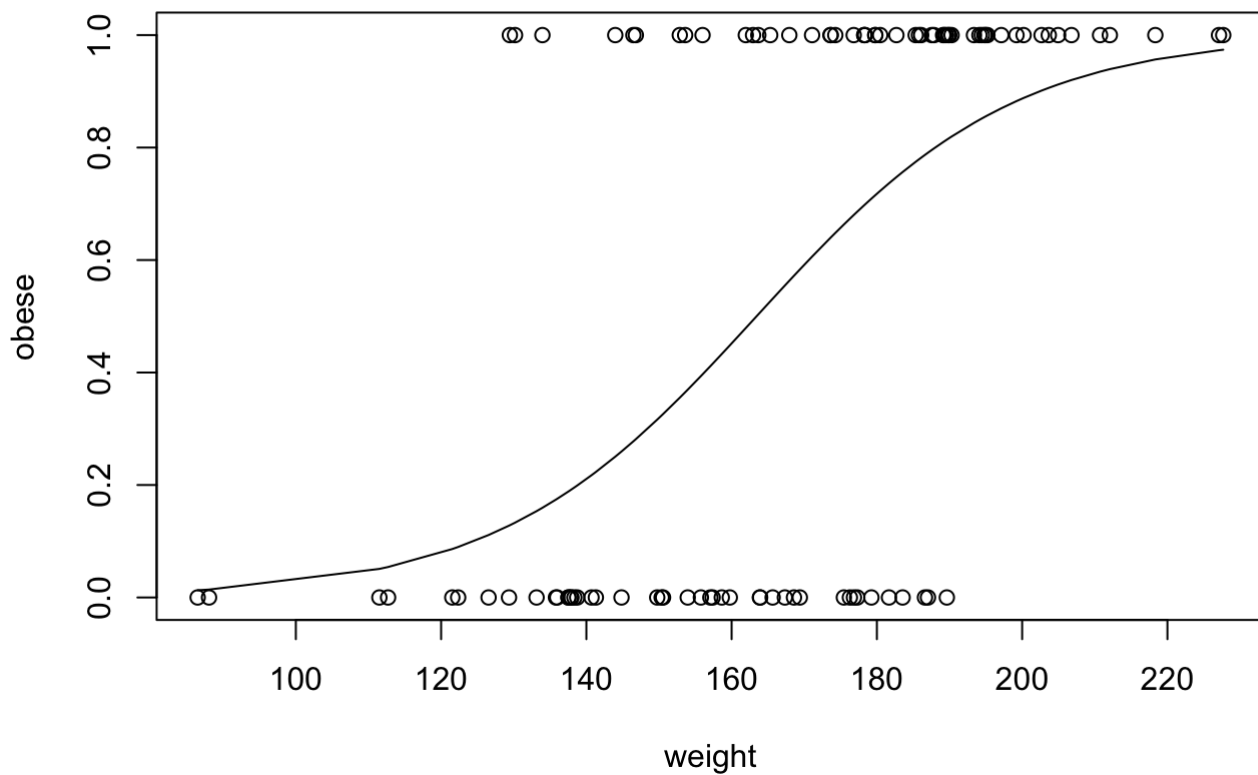
```
## randomForest 4.6-14
```

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

```
set.seed(420)  
  
num.samples <- 100  
  
weight <- sort(rnorm(n=num.samples, mean=172, sd=29))  
  
obese <- ifelse(test=(runif(n=num.samples) < (rank(weight)/num.samples)),  
  yes=1, no=0)  
obese
```

```
##      [1] 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 1 1 0 0 1 0 0 0  
##      [38] 0 1 1 1 0 0 1 0 0 1 0 0 1 1 1 1 0 0 1 0 0 1 1 1 0 1 1 1 0 1 0 1 1 1 0 0 1  
##      [75] 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
plot(x=weight, y=obese)  
  
glm.fit=glm(obese ~ weight, family=binomial)  
lines(weight, glm.fit$fitted.values)
```



```
roc(obese, glm.fit$fitted.values, plot=TRUE)
```

```
## Setting levels: control = 0, case = 1
```

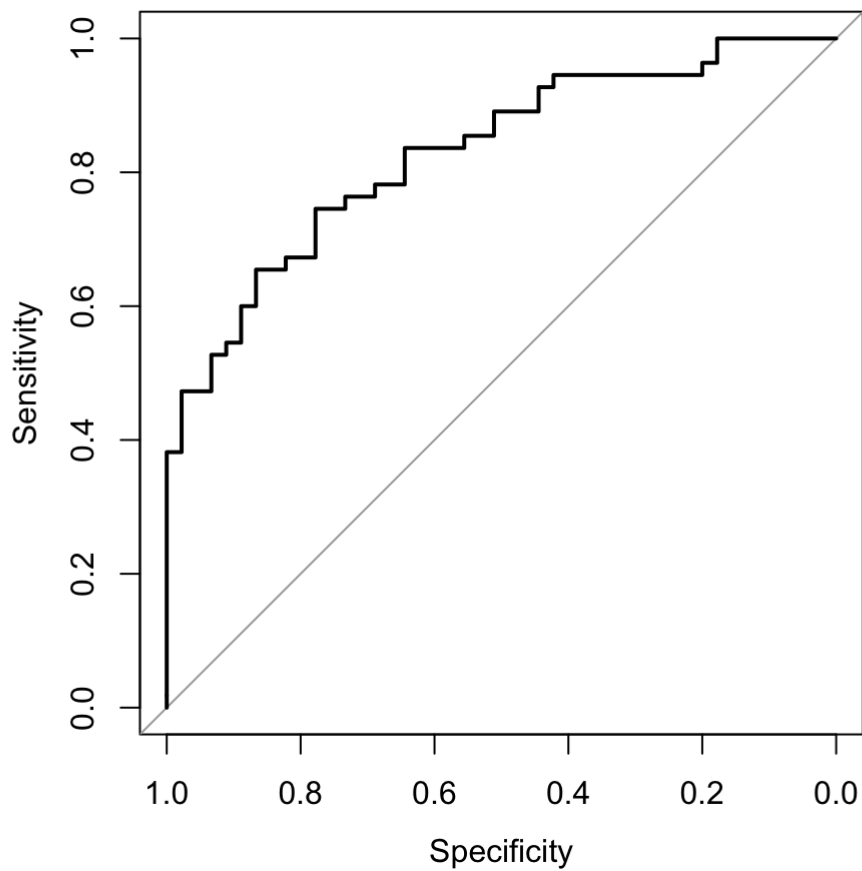
```
## Setting direction: controls < cases
```

```
##
## Call:
## roc.default(response = obese, predictor = glm.fit$fitted.values,      plot = TRUE)
##
## Data: glm.fit$fitted.values in 45 controls (obese 0) < 55 cases (obese 1).
## Area under the curve: 0.8291
```

```
par(pty = "s")
```

```
roc(obese, glm.fit$fitted.values, plot=TRUE)
```

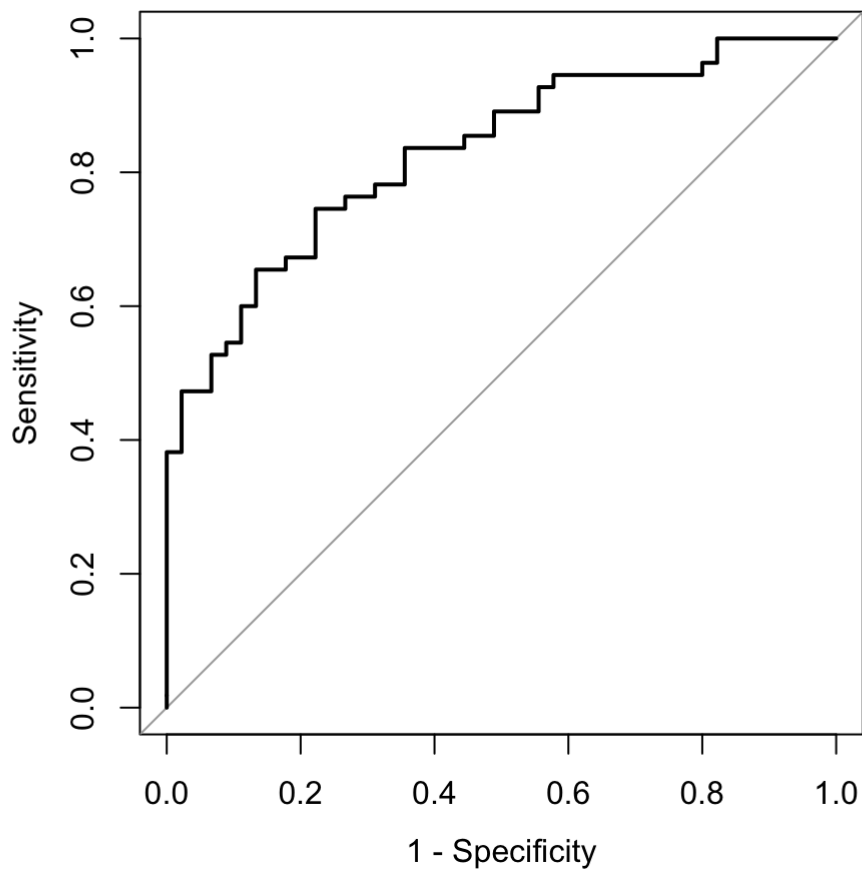
```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```



```
##  
## Call:  
## roc.default(response = obese, predictor = glm.fit$fitted.values,      plot = TRUE)  
##  
## Data: glm.fit$fitted.values in 45 controls (obese 0) < 55 cases (obese 1).  
## Area under the curve: 0.8291
```

```
roc(obese, glm.fit$fitted.values, plot=TRUE, legacy.axes=TRUE)
```

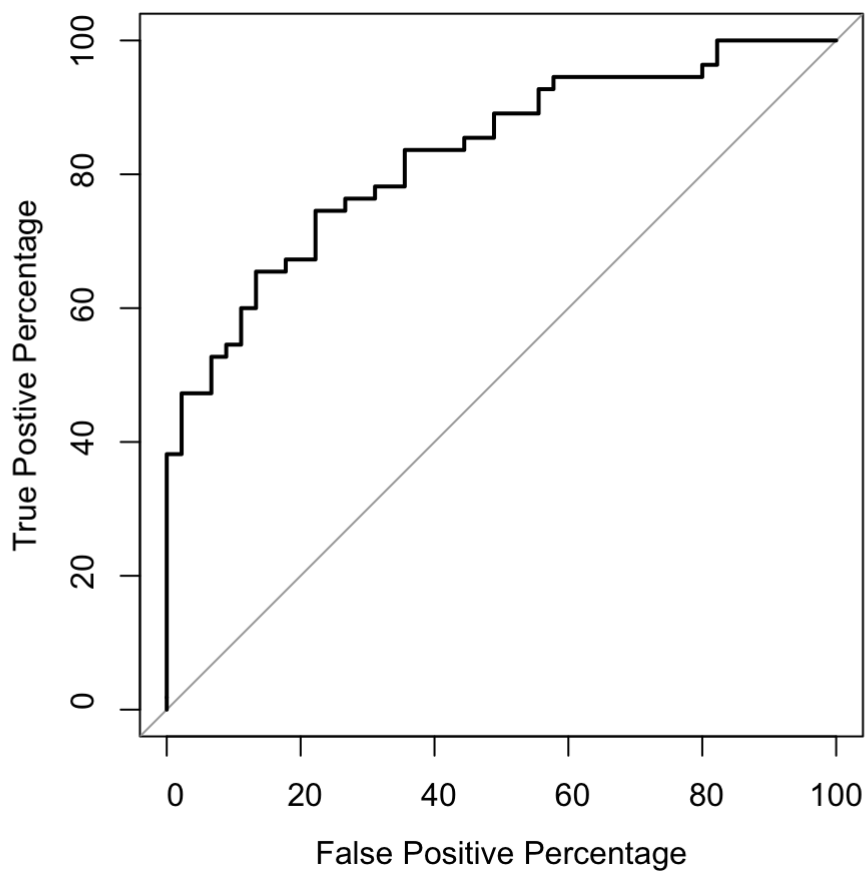
```
## Setting levels: control = 0, case = 1  
## Setting direction: controls < cases
```



```
##  
## Call:  
## roc.default(response = obese, predictor = glm.fit$fitted.values,      plot = TRUE,  
## legacy.axes = TRUE)  
##  
## Data: glm.fit$fitted.values in 45 controls (obese 0) < 55 cases (obese 1).  
## Area under the curve: 0.8291
```

```
roc(obese, glm.fit$fitted.values, plot=TRUE, legacy.axes=TRUE, percent=TRUE, xlab="Fa  
lse Positive Percentage", ylab="True Postive Percentage")
```

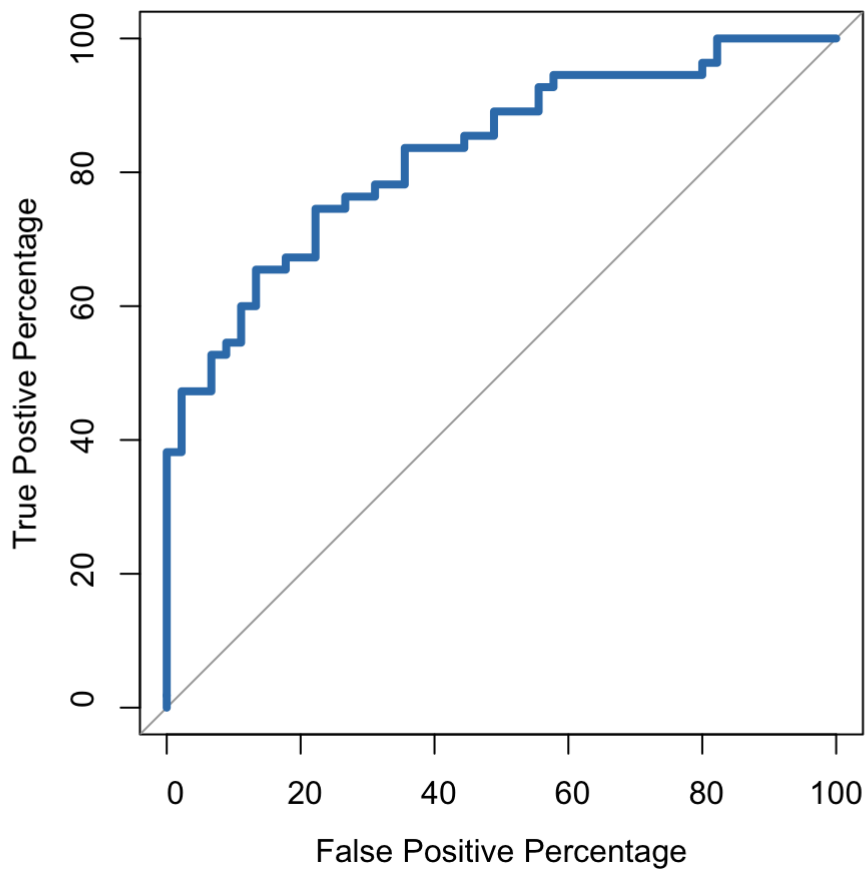
```
## Setting levels: control = 0, case = 1  
## Setting direction: controls < cases
```



```
##
## Call:
## roc.default(response = obese, predictor = glm.fit$fitted.values, percent = TRUE,
## E, plot = TRUE, legacy.axes = TRUE, xlab = "False Positive Percentage", ylab = "True
## Postive Percentage")
##
## Data: glm.fit$fitted.values in 45 controls (obese 0) < 55 cases (obese 1).
## Area under the curve: 82.91%
```

```
roc(obese, glm.fit$fitted.values, plot=TRUE, legacy.axes=TRUE, percent=TRUE, xlab="False
Positive Percentage", ylab="True Postive Percentage", col="#377eb8", lwd=4)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```



```
##
## Call:
## roc.default(response = obese, predictor = glm.fit$fitted.values, percent = TRUE,
## E, plot = TRUE, legacy.axes = TRUE, xlab = "False Positive Percentage", ylab = "True
## rue Postive Percentage", col = "#377eb8", lwd = 4)
##
## Data: glm.fit$fitted.values in 45 controls (obese 0) < 55 cases (obese 1).
## Area under the curve: 82.91%
```

```
roc.info <- roc(obese, glm.fit$fitted.values, legacy.axes=TRUE)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
str(roc.info)
```

```

## List of 15
## $ percent          : logi FALSE
## $ sensitivities     : num [1:101] 1 1 1 1 1 ...
## $ specificities     : num [1:101] 0 0.0222 0.0444 0.0667 0.0889 ...
## $ thresholds        : num [1:101] -Inf 0.0135 0.0325 0.0525 0.0702 ...
## $ direction         : chr "<"
## $ cases             : Named num [1:55] 0.128 0.133 0.159 0.25 0.278 ...
## ..- attr(*, "names")= chr [1:55] "9" "10" "12" "23" ...
## $ controls          : Named num [1:45] 0.0129 0.0141 0.0508 0.0542 0.0862 ...
## ..- attr(*, "names")= chr [1:45] "1" "2" "3" "4" ...
## $ fun.sesp          :function (thresholds, controls, cases, direction)
## $ auc               : 'auc' num 0.829
## ..- attr(*, "partial.auc")= logi FALSE
## ..- attr(*, "percent")= logi FALSE
## ..- attr(*, "roc")=List of 15
## .. ..$ percent      : logi FALSE
## .. ..$ sensitivities : num [1:101] 1 1 1 1 1 ...
## .. ..$ specificities : num [1:101] 0 0.0222 0.0444 0.0667 0.0889 ...
## .. ..$ thresholds    : num [1:101] -Inf 0.0135 0.0325 0.0525 0.0702 ...
## .. ..$ direction     : chr "<"
## .. ..$ cases         : Named num [1:55] 0.128 0.133 0.159 0.25 0.278 ...
## .. ..- attr(*, "names")= chr [1:55] "9" "10" "12" "23" ...
## .. ..$ controls      : Named num [1:45] 0.0129 0.0141 0.0508 0.0542 0.0862
## ...
## .. ..- attr(*, "names")= chr [1:45] "1" "2" "3" "4" ...
## .. ..$ fun.sesp      :function (thresholds, controls, cases, direction)
## .. ..$ auc           : 'auc' num 0.829
## .. ..- attr(*, "partial.auc")= logi FALSE
## .. ..- attr(*, "percent")= logi FALSE
## .. ..- attr(*, "roc")=List of 8
## .. .. ..$ percent    : logi FALSE
## .. .. ..$ sensitivities: num [1:101] 1 1 1 1 1 ...
## .. .. ..$ specificities: num [1:101] 0 0.0222 0.0444 0.0667 0.0889 ...
## .. .. ..$ thresholds  : num [1:101] -Inf 0.0135 0.0325 0.0525 0.0702 ...
## .. .. ..$ direction   : chr "<"
## .. .. ..$ cases      : Named num [1:55] 0.128 0.133 0.159 0.25 0.278 ...
## .. .. ..- attr(*, "names")= chr [1:55] "9" "10" "12" "23" ...
## .. .. ..$ controls    : Named num [1:45] 0.0129 0.0141 0.0508 0.0542 0.0862
## ...
## .. .. ..- attr(*, "names")= chr [1:45] "1" "2" "3" "4" ...
## .. .. ..$ fun.sesp    :function (thresholds, controls, cases, direction)
## .. .. ..- attr(*, "class")= chr "roc"
## .. ..$ call           : language roc.default(response = obese, predictor = glm.fit$fitted.values, legacy.axes = TRUE)
## .. ..$ original.predictor: Named num [1:100] 0.0129 0.0141 0.0508 0.0542 0.0862
## ...
## .. ..- attr(*, "names")= chr [1:100] "1" "2" "3" "4" ...
## .. ..$ original.response : num [1:100] 0 0 0 0 0 0 0 0 1 1 ...
## .. ..$ predictor         : Named num [1:100] 0.0129 0.0141 0.0508 0.0542 0.0862
## ...
## .. ..- attr(*, "names")= chr [1:100] "1" "2" "3" "4" ...
## .. ..$ response         : num [1:100] 0 0 0 0 0 0 0 0 1 1 ...
## .. ..$ levels           : chr [1:2] "0" "1"
## .. ..- attr(*, "class")= chr "roc"
## $ call                  : language roc.default(response = obese, predictor = glm.fit
$fitted.values, legacy.axes = TRUE)
## $ original.predictor: Named num [1:100] 0.0129 0.0141 0.0508 0.0542 0.0862 ...

```

```
##    ..- attr(*, "names")= chr [1:100] "1" "2" "3" "4" ...
##    $ original.response : num [1:100] 0 0 0 0 0 0 0 0 1 1 ...
##    $ predictor          : Named num [1:100] 0.0129 0.0141 0.0508 0.0542 0.0862 ...
##    ..- attr(*, "names")= chr [1:100] "1" "2" "3" "4" ...
##    $ response           : num [1:100] 0 0 0 0 0 0 0 0 1 1 ...
##    $ levels             : chr [1:2] "0" "1"
##    - attr(*, "class")= chr "roc"
```

```
roc.df <- data.frame(
  tpp=roc.info$sensitivities*100,
  fpp=(1 - roc.info$specificities)*100,
  thresholds=roc.info$thresholds)
```

```
head(roc.df)
```

```
##    tpp      fpp thresholds
## 1 100 100.00000      -Inf
## 2 100  97.77778 0.01349011
## 3 100  95.55556 0.03245008
## 4 100  93.33333 0.05250145
## 5 100  91.11111 0.07017225
## 6 100  88.88889 0.08798755
```

```
tail(roc.df)
```

```
##          tpp fpp thresholds
## 96  9.090909  0  0.9275222
## 97  7.272727  0  0.9371857
## 98  5.454545  0  0.9480358
## 99  3.636364  0  0.9648800
## 100 1.818182  0  0.9735257
## 101 0.000000  0          Inf
```

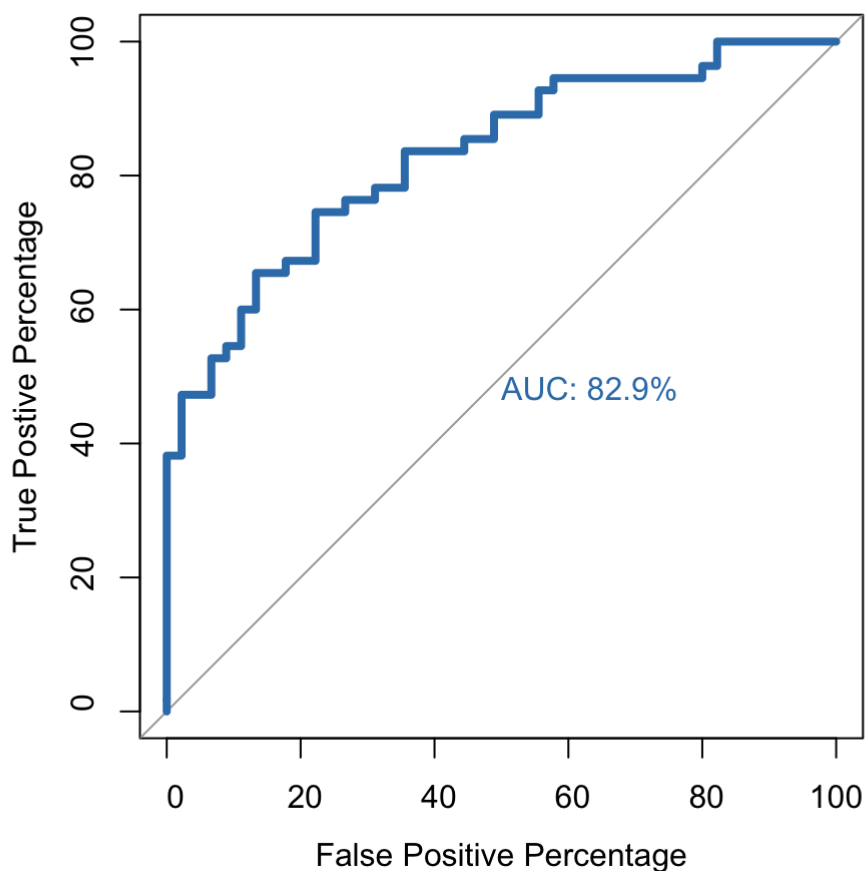
```
roc.df[roc.df$tpp > 60 & roc.df$tpp < 80,]
```



```
##          tpp          fpp thresholds
## 42 78.18182 35.55556 0.5049310
## 43 78.18182 33.33333 0.5067116
## 44 78.18182 31.11111 0.5166680
## 45 76.36364 31.11111 0.5287933
## 46 76.36364 28.88889 0.5429351
## 47 76.36364 26.66667 0.5589494
## 48 74.54545 26.66667 0.5676342
## 49 74.54545 24.44444 0.5776086
## 50 74.54545 22.22222 0.5946054
## 51 72.72727 22.22222 0.6227449
## 52 70.90909 22.22222 0.6398136
## 53 69.09091 22.22222 0.6441654
## 54 67.27273 22.22222 0.6556705
## 55 67.27273 20.00000 0.6683618
## 56 67.27273 17.77778 0.6767661
## 57 65.45455 17.77778 0.6802060
## 58 65.45455 15.55556 0.6831936
## 59 65.45455 13.33333 0.6917225
## 60 63.63636 13.33333 0.6975300
## 61 61.81818 13.33333 0.6982807
```

```
roc(obese, glm.fit$fitted.values, plot=TRUE, legacy.axes=TRUE, percent=TRUE, xlab="False Positive Percentage", ylab="True Positive Percentage", col="#377eb8", lwd=4, print.auc=TRUE)
```

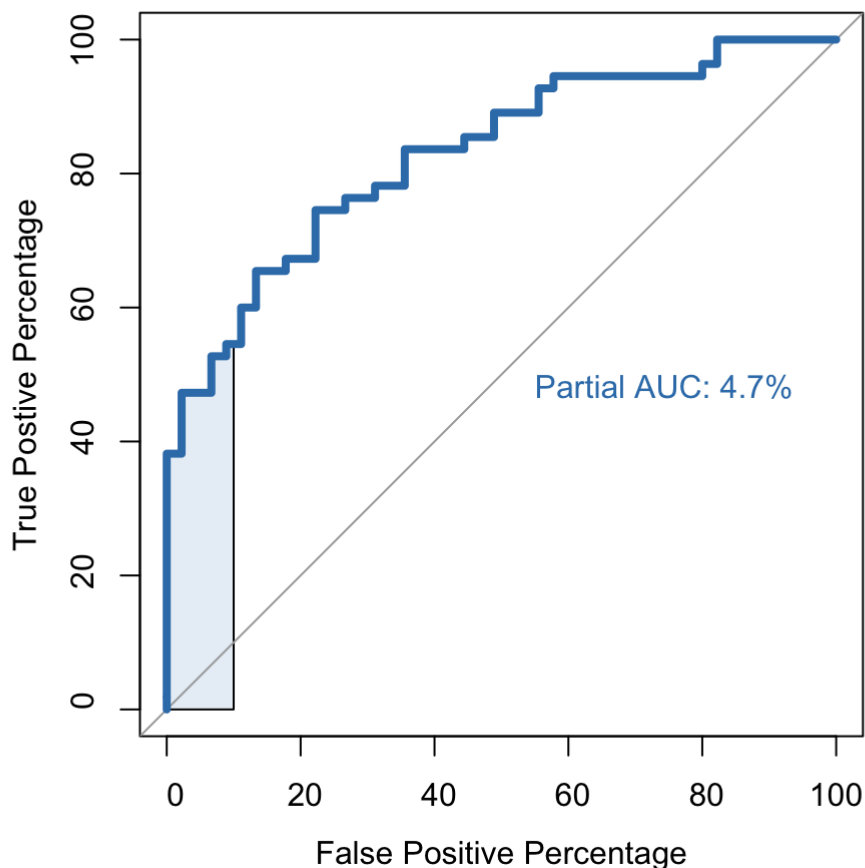
```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```



```
##
## Call:
## roc.default(response = obese, predictor = glm.fit$fitted.values, percent = TRUE,
## plot = TRUE, legacy.axes = TRUE, xlab = "False Positive Percentage", ylab = "True
## Postive Percentage", col = "#377eb8", lwd = 4, print.auc = TRUE)
##
## Data: glm.fit$fitted.values in 45 controls (obese 0) < 55 cases (obese 1).
## Area under the curve: 82.91%
```

```
roc(obese, glm.fit$fitted.values, plot=TRUE, legacy.axes=TRUE, percent=TRUE, xlab="False
Positive Percentage", ylab="True Postive Percentage", col="#377eb8", lwd=4, print.auc=TRUE,
print.auc.x=45, partial.auc=c(100, 90), auc.polygon = TRUE, auc.polygon.col = "#377eb822")
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```



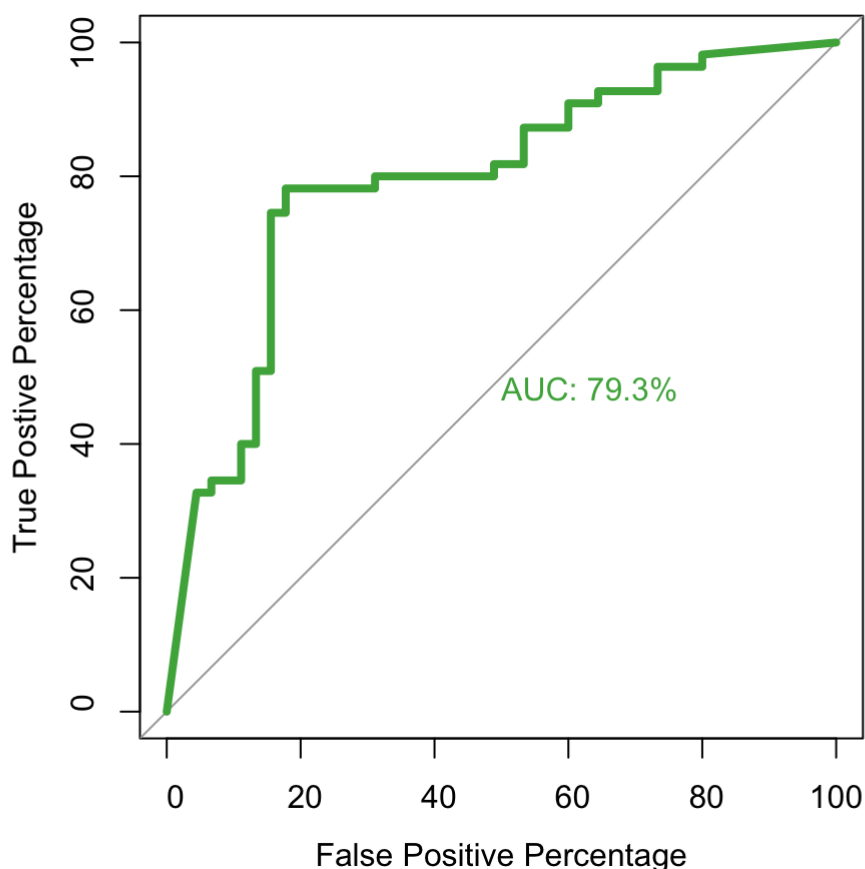
```
##
## Call:
## roc.default(response = obese, predictor = glm.fit$fitted.values, percent = TRUE,
## plot = TRUE, legacy.axes = TRUE, xlab = "False Positive Percentage", ylab = "True
## Postive Percentage", col = "#377eb8", lwd = 4, print.auc = TRUE, print.auc.x
## = 45, partial.auc = c(100, 90), auc.polygon = TRUE, auc.polygon.col = "#377eb
## 822")
##
## Data: glm.fit$fitted.values in 45 controls (obese 0) < 55 cases (obese 1).
## Partial area under the curve (specificity 100%-90%): 4.727%
```

```
rf.model <- randomForest(factor(obese) ~ weight)

roc(obese, rf.model$votes[,1], plot=TRUE, legacy.axes=TRUE, percent=TRUE, xlab="False
Positive Percentage", ylab="True Postive Percentage", col="#4daf4a", lwd=4, print.auc
=TRUE)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls > cases
```



```
##
## Call:
## roc.default(response = obese, predictor = rf.model$votes[, 1], percent = TRUE,
plot = TRUE, legacy.axes = TRUE, xlab = "False Positive Percentage", ylab = "True
Postive Percentage", col = "#4daf4a", lwd = 4, print.auc = TRUE)
##
## Data: rf.model$votes[, 1] in 45 controls (obese 0) > 55 cases (obese 1).
## Area under the curve: 79.29%
```

```
roc(obese, glm.fit$fitted.values, plot=TRUE, legacy.axes=TRUE, percent=TRUE, xlab="Fa
lse Positive Percentage", ylab="True Postive Percentage", col="#377eb8", lwd=4, prin
t.auc=TRUE)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

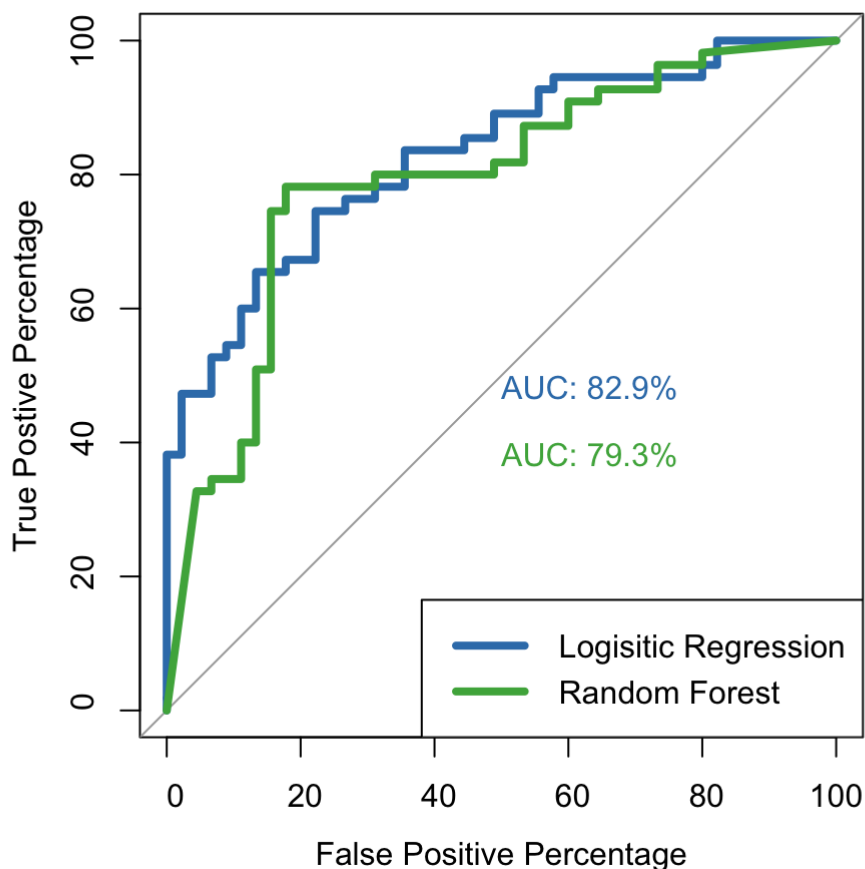
```
##
## Call:
## roc.default(response = obese, predictor = glm.fit$fitted.values, percent = TRUE,
## plot = TRUE, legacy.axes = TRUE, xlab = "False Positive Percentage", ylab = "True
## Positive Percentage", col = "#377eb8", lwd = 4, print.auc = TRUE)
##
## Data: glm.fit$fitted.values in 45 controls (obese 0) < 55 cases (obese 1).
## Area under the curve: 82.91%
```

```
plot.roc(obese, rf.model$votes[,1], percent=TRUE, col="#4daf4a", lwd=4, print.auc=TRUE,
add=TRUE, print.auc.y=40)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls > cases
```

```
legend("bottomright", legend=c("Logisitic Regression", "Random Forest"), col=c("#377eb8",
"#4daf4a"), lwd=4)
```



```
par(pty = "m")
```