Xkong_HW2

Vivian Kong

6/16/2018

1. Read data

```
score_class<- read.csv("https://raw.githubusercontent.com/xkong100/data-621/master/HW2/classification-ou
head(score_class)
```

```
##   pregnant glucose diastolic skinfold insulin  bmi pedigree age class
## 1        7     124        70       33     215 25.5    0.161  37     0
## 2        2     122        76       27     200 35.9    0.483  26     0
## 3        3     107        62       13      48 22.9    0.678  23     1
## 4        1      91        64       24       0 29.2    0.192  21     0
## 5        4      83        86       19       0 29.3    0.317  34     0
## 6        1     100        74       12      46 19.5    0.149  28     0
##   scored.class scored.probability
## 1            0         0.32845226
## 2            0         0.27319044
## 3            0         0.10966039
## 4            0         0.05599835
## 5            0         0.10049072
## 6            0         0.05515460
```

2. confusion Matrix

```
t <- table(score_class$class, score_class$scored.class)
colnames(t) <- c("Real Negative", "Real Positive")
rownames(t) <- c("Model Negative", "Model Positive")
t
```

```
##
##                  Real Negative Real Positive
##   Model Negative           119             5
##   Model Positive            30            27
```

The columns represent for Scored.Class which represent the predicted classes, The rows represents for the class which represent the predicted classes. “1=positive”, “0=negative”

3. Accuracy

```
accuracy <- function(data){
  t <- table(data$class, data$scored.class)
  tn<-t[1,1]
  tp<-t[2,2]
  fn<-t[2,1]
  fp<-t[1,2]
acc <- (tp+tn)/ (tp+fp+tn+fn)

return (acc)

}
a<-accuracy(score_class)
a
```

```
## [1] 0.8066298
```

4. Classification Error Rate

```
error_rate <- function(data){
  t <- table(data$class, data$scored.class)
  tn<-t[1,1]
  tp<-t[2,2]
  fn<-t[2,1]
  fp<-t[1,2]
error <- (fp+fn)/ (tp+fp+tn+fn)

return (error)

}
e<-error_rate(score_class)
e
```

```
## [1] 0.1933702
```

```
sum<-accuracy(score_class)+error_rate(score_class)
sum
```

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

5. Precision

```
precision <- function(data){
  t <- table(data$class, data$scored.class)
  tn<-t[1,1]
  tp<-t[2,2]
  fn<-t[2,1]
  fp<-t[1,2]
pre <- (tp)/ (tp+fp)

return (pre)

}
pre<-precision(score_class)
pre
```

```
## [1] 0.84375
```

6. Sensitivity

```
sensitivity <- function(data){
  t <- as.data.frame(table(Actual=data$class,Predicted= data$scored.class))
return(t$Freq[4]/(t$Freq[4]+t$Freq[2]))

}
sen<-sensitivity(score_class)
sen
```

```
## [1] 0.4736842
```

7. Specificity

```
specificity <- function(data){
  t <- as.data.frame(table(Actual=data$class,Predicted= data$scored.class))
return(t$Freq[1]/(t$Freq[1]+t$Freq[3]))
```

```
}
sp<-specificity(score_class)
sp
```

```
## [1] 0.9596774
```

8. F1 Score

```
F1_Score <- function(data){
   t <- table(data$class, data$scored.class)
  tn<-t[1,1]
  tp<-t[2,2]
  fn<-t[2,1]
  fp<-t[1,2]

  pre <- (tp)/ (tp+fp)
  sen <- (tp)/ (tp+fn)
  F1 <- (2*pre*sen)/(pre+sen)
  return(F1)
}
F1<- F1_Score(score_class)
F1
```

```
## [1] 0.6067416
```

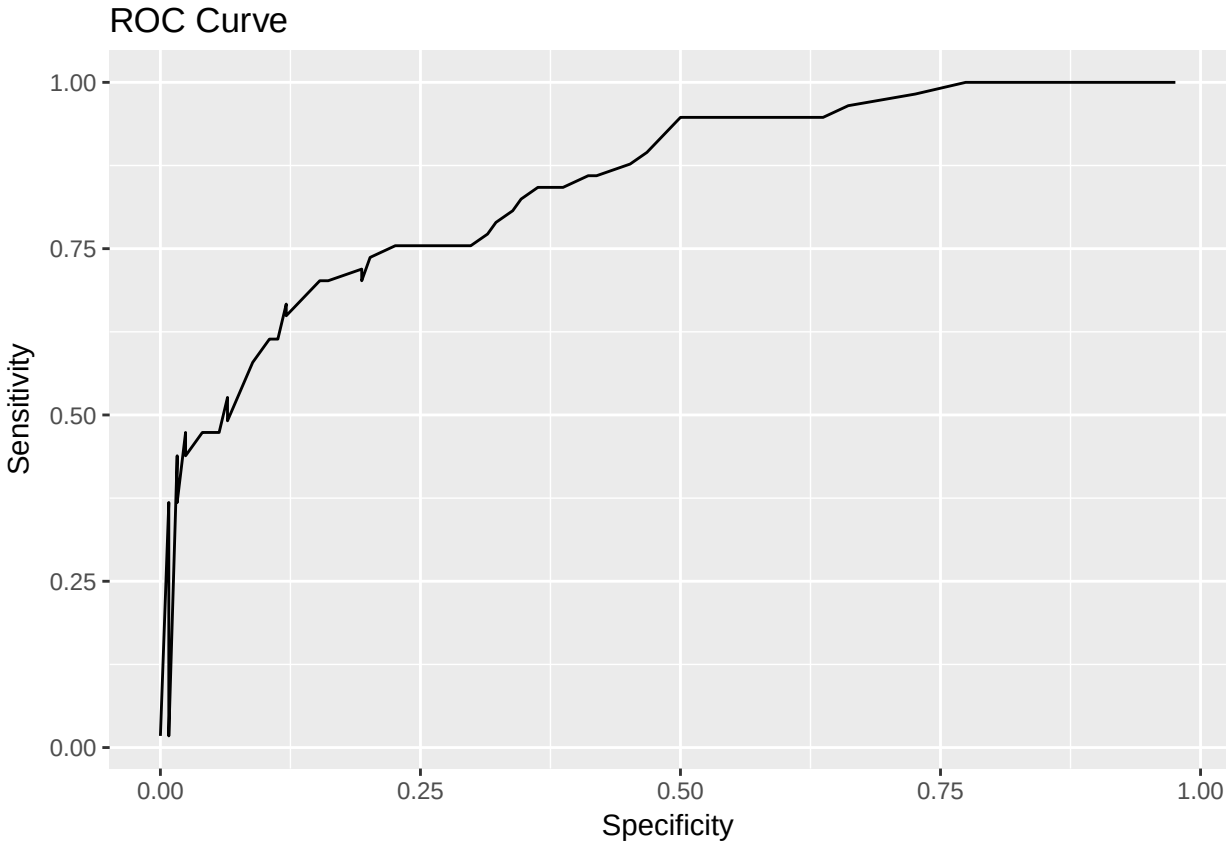
9. Let $P = Precision$, $S = Sensitivity$, Since $0 < p < 1$ and $0 < s < 1$, we know that $p(s-1) < 0$ *and* $s(p-1) < 0$. In this case, we can find that $ps < p,\ sp < s.\ ps + sp < p + s,\ 2ps < p + s,\ \frac{2ps}{p+s} < 1.$ Since $p > 0,\ s > 0,\ \frac{2ps}{p+s} > 0.$

In conclusion, $0 < F1Score < 1$.

10. ROC Curve and AUC curve

```
library(ggplot2)
ROC <- function(data)
{
  data1 = data
  thresholds <- seq(0,1,0.01)
  Y <- c()
  X <- c()
  for (threshod in thresholds) {
    data1$scored.class <- ifelse(data1$scored.probability > threshod,1,0)
    X <- append(X,1-specificity(data1))
    Y <- append(Y,sensitivity(data1))
    }
  df <- data.frame(X=X,Y=Y)
  df <- na.omit(df)
  g <- ggplot(df,aes(X,Y)) + geom_line() + ggtitle('ROC Curve') +
    xlab('Specificity') + ylab('Sensitivity')
  height = (df$Y[-1]+df$Y[-length(df$Y)])/2
  width = -diff(df$X)
  area = sum(height*width)
  return(list(Plot =g,AUC = area))
}
r <-ROC(score_class)
r
```

```
## $Plot
```




```
##
## $AUC
## [1] 0.8247029
```

11. Matrix

```
list(accuracy=a, error_rate=e, precision=pre, sensitivity=sen, specificity=sp, F1=F1, Auc= r$AUC)
```

```
## $accuracy
## [1] 0.8066298
##
## $error_rate
## [1] 0.1933702
##
## $precision
## [1] 0.84375
##
## $sensitivity
## [1] 0.4736842
##
## $specificity
## [1] 0.9596774
##
## $F1
## [1] 0.6067416
##
## $Auc
```

```
## [1] 0.8247029
```

12. Investigate the "Caret" package

```
library(caret)
```

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

```
##
## Attaching package: 'caret'
```

```
## The following objects are masked _by_ '.GlobalEnv':
##
##     precision, sensitivity, specificity
```

```
confusionMatrix(as.factor(score_class$scored.class), as.factor(score_class$class), positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 119  30
##          1   5  27
##
##                Accuracy : 0.8066
##                  95% CI : (0.7415, 0.8615)
##     No Information Rate : 0.6851
##     P-Value [Acc > NIR] : 0.0001712
##
##                   Kappa : 0.4916
##  Mcnemar's Test P-Value : 4.976e-05
##
##             Sensitivity : 0.4737
##             Specificity : 0.9597
##          Pos Pred Value : 0.8438
##          Neg Pred Value : 0.7987
##              Prevalence : 0.3149
##          Detection Rate : 0.1492
##    Detection Prevalence : 0.1768
##       Balanced Accuracy : 0.7167
##
##        'Positive' Class : 1
##
```

13. Investigate the "pROC" package

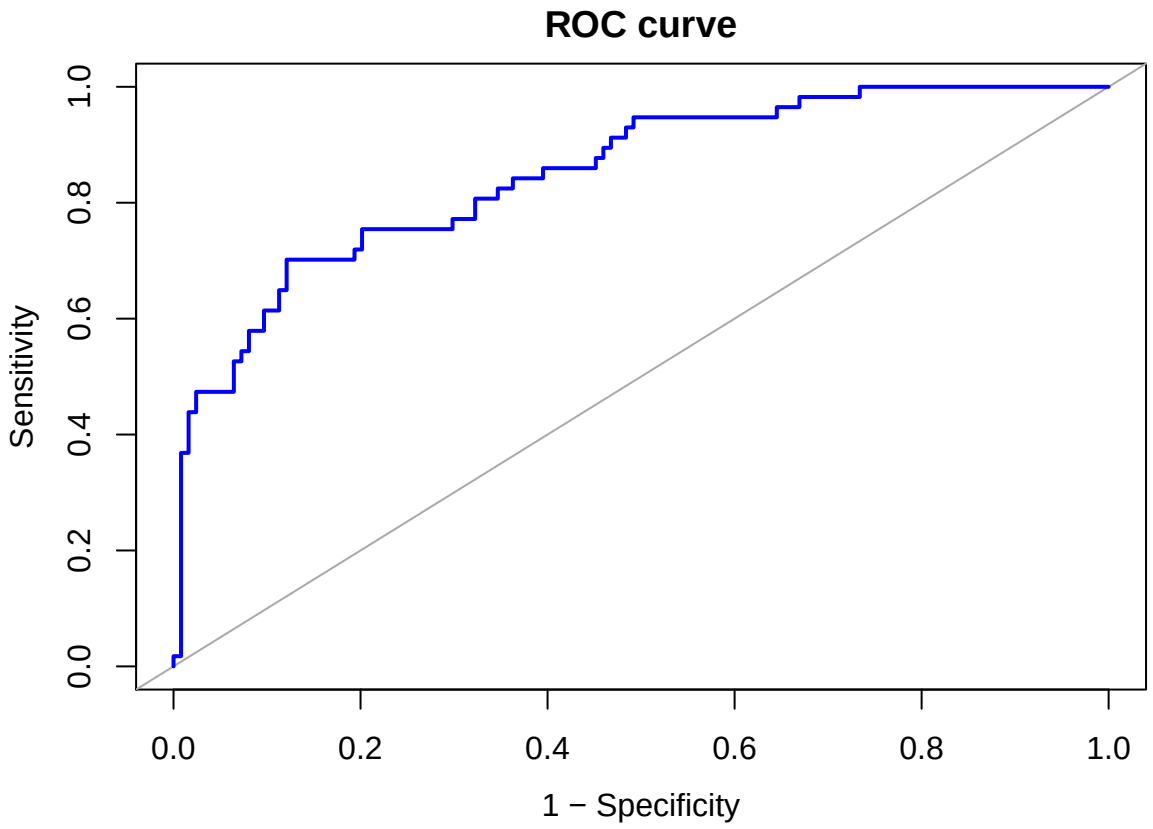
```
library(pROC)
```

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

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

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

```
roc <- roc(score_class$class, score_class$scored.probability, plot=T, asp=NA, legacy.axes=T, main="ROC
```




```
roc["auc"]
```

```
## $auc
## Area under the curve: 0.8503
```