Discriminant Analysis

```
> head(train_numeric,5)
  Age Rank avgOdds SP_Percent BP_Win_Percentage Aces firstServeReturnsWon SecondServeRetu
1  21  129    0.00  0.5777778         0.4166667    2                   10
2  23   74    1.57  0.5894737         0.7500000    4                   24
3  23   74    4.12  0.6896552         0.6363636    5                   15
4  24   91    8.33  0.5175439         0.2500000    7                   20
5  25  170    9.58  0.7435897         0.4000000    5                    7
  FirstServesIn DoubleFaults FirstServePercentage winPercentage Finalists
1            27            4             0.4354839     0.0000000         0
2            47            3             0.5949367     0.3333333         0
3            64            3             0.5981308     0.3333333         0
4            69            3             0.6831683     0.2500000         0
5            36            5             0.5901639     0.2000000         0
> head(test_numeric,5)
  Age Rank avgOdds SP_Percent BP_Win_Percentage Aces firstServeReturnsWon SecondServeRetu
1  22   86    1.43  0.6300000         0.3333333   13                   16
2  26   71    2.10  0.6267606         0.3333333    3                   19
3  26   71   10.70  0.6106195         0.4705882    3                   25
4  27   36    1.25  0.5982906         0.8888889    6                   30
5  27   36    1.83  0.5283019         0.5263158    5                   24
  FirstServesIn DoubleFaults FirstServePercentage winPercentage Finalists
1            66            3             0.6168224     0.0000000         0
2            81            3             0.5785714     0.2857143         0
3            77            1             0.5620438     0.2857143         0
4            58            5             0.5631068     0.3333333         0
5            68           10             0.6355140     0.3333333         0


>
> train.lda <- lda( Finalists~., data = train_numeric)
> summary(train.lda)
        Length Class  Mode
prior    2     -none- numeric
counts   2     -none- numeric
means   24     -none- numeric
scaling 12     -none- numeric
lev      2     -none- character
svd      1     -none- numeric
N        1     -none- numeric
call     3     -none- call
terms    3     terms  call
xlevels  0     -none- list
> print(train.lda)
Call:
lda(Finalists ~ ., data = train_numeric)

Prior probabilities of groups:
         0          1
0.94497743 0.05502257

Group means:
        Age     Rank   avgOdds SP_Percent BP_Win_Percentage     Aces firstServeReturnsWon
0 25.75754 70.62377 0.8282084  0.6263172         0.5650589 8.106599             19.60794
1 25.21026 11.23077 0.2817949  0.5913228         0.5665052 9.558974             21.80000
  SecondServeReturnsWon FirstServesIn DoubleFaults FirstServePercentage winPercentage
0              21.78979      66.94416     4.170200            0.6027583     0.4722776
1              23.85128      60.94359     2.394872            0.6159582     0.8276409
```

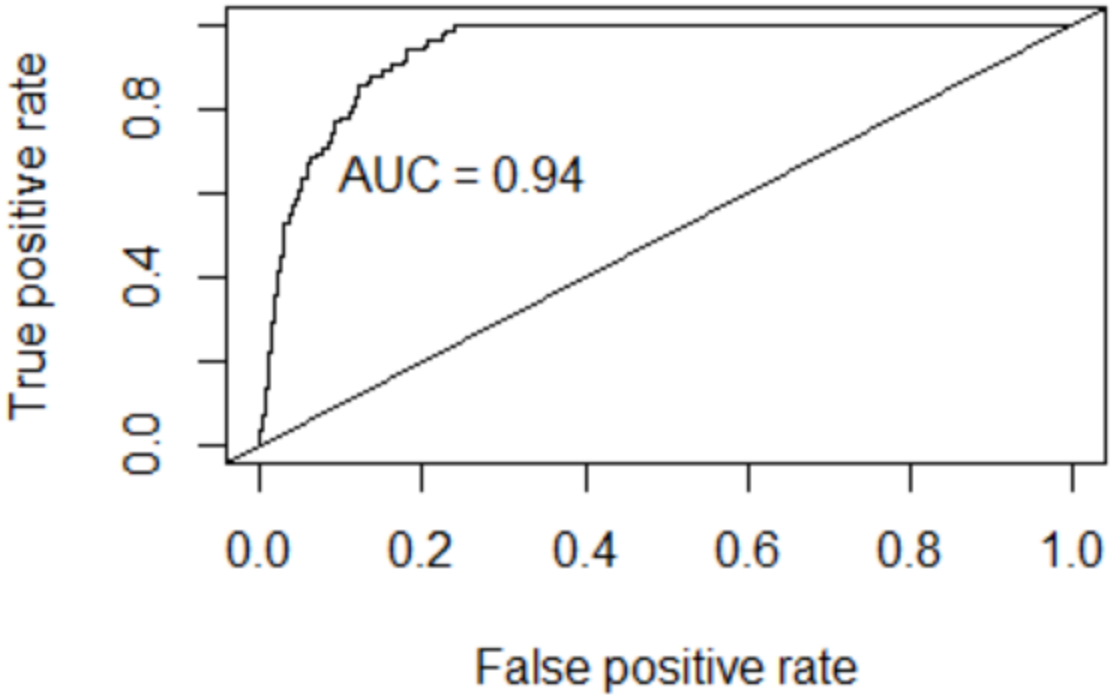
```
Coefficients of linear discriminants:
                              LD1
Age                   -0.036516073
Rank                  -0.002064262
avgOdds               -0.007786522
SP_Percent            -0.514616517
BP_Win_Percentage     -0.284386135
Aces                   0.015339455
firstServeReturnsWon   0.008697735
SecondServeReturnsWon  0.011585482
FirstServesIn         -0.008800316
DoubleFaults          -0.078235182
FirstServePercentage   0.746675172
winPercentage          3.257525606
> plot(train.lda)
Error in plot.new() : figure margins too large


>
train.lda$counts
   0    1
3349  195
> train.lda$means
       Age     Rank   avgOdds SP_Percent BP_Win_Percentage     Aces firstServ
eReturnsWon
0 25.75754 70.62377 0.8282084  0.6263172         0.5650589 8.106599
19.60794
1 25.21026 11.23077 0.2817949  0.5913228         0.5665052 9.558974
21.80000
  SecondServeReturnsWon FirstServesIn DoubleFaults FirstServePercentage winPe
rcentage
0              21.78979      66.94416     4.170200            0.6027583     0
.4722776
1              23.85128      60.94359     2.394872            0.6159582     0
.8276409
> train.lda$scaling
                              LD1
Age                   -0.036516073
Rank                  -0.002064262
avgOdds               -0.007786522
SP_Percent            -0.514616517
BP_Win_Percentage     -0.284386135
Aces                   0.015339455
firstServeReturnsWon   0.008697735
SecondServeReturnsWon  0.011585482
FirstServesIn         -0.008800316
DoubleFaults          -0.078235182
FirstServePercentage   0.746675172
winPercentage          3.257525606
> train.lda$prior
         0          1
0.94497743 0.05502257
> train.lda$lev
[1] "0" "1"
> train.lda$svd
[1] 21.56776


> lda.predict <- predict(train.lda, newdata = test_numeric)
> lda.predict$class
   [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
```

```
  [47] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
  [93] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
 [139] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
 [185] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
 [231] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
 [277] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
 [323] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
 [369] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
 [415] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
 [461] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
 [507] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
 [553] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
 [599] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
 [645] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
 [691] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
 [737] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
 [783] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
 [829] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
 [875] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
 [921] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
 [967] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [ reached getOption("max.print") -- omitted 504 entries ]
Levels: 0 1
```

```
> library(ROCR)
> red <- prediction(lda.predict.posteriors[,2], test_numeric$Finalists)
> roc.perf = performance(pred, measure = "tpr", x.measure = "fpr")
> auc.train <- performance(pred, measure = "auc")
> auc.train <- auc.train@y.values
> plot(roc.perf)
> abline(a=0, b= 1)
>
> text(x = .25, y = .65 ,paste("AUC = ", round(auc.train[[1]],3), sep = ""))
```


True positive rate
0.0
0.4
0.8
AUC = 0.94
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate