04.Analise Discriminante e Arvore de Decisao

Felipe Rocha

2022-12-17

Aula 07: Análise Discriminante

Antes de tudo, carregando os pacotes

```
#Instalar Pacote
#install.packages("MASS")
#install.packages("tree")
#install.packages("caret")

library(MASS) #QDA Analise de discriminante quadratica e LDA Linear Discriminant Analysis
library(tree) #Arvore de decisao
library(caret) #Medidas de Performance
library(randomForest) #Bagging e randomForest
```

Iris

Validação Cruzada (leave one out)

Para realizar essa anlise, vamos utilizar o procedimento de validação cruzada, conhecida como *leave one out*. Esse procedimento é utilizado para comparar a capacidade preditiva do modelo - Por exemplo, se temos um conjunto de dados com 300 indivíduos, estimamos a função com 299 e predizemos a função para o próximo índivudo, e assim fazemos de 1 em 1. Esse procedimento é utilizado para evitar a superestimação do poder de classificação do modelo, ou seja, para tentar fazer a predição de maneira mais independente possível.

```
r <- lda(formula = Species ~ ., #especi em funcao de todas as outras vars
         data = iris, #do dados iris
         prior = c(1,1,1)/3, #tenho 3 populacoes e defini probabilidade de
         #ocorrencia de cada populacao iguais para todas (1/3)
         CV=TRUE) #CV=TRUE: quero validacao cruzada (leave one out)
```

Funçaõ Linear Construindo a tabela de confusão, onde o número de classificações corretas ficam na diagonal.

```
# Funcao Linear
classificacao <- r$class
cvl <- table(iris$Species,classificacao)
cvl
```

```
##             classificacao
##              setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         48         2
##   virginica       0          1        49
```

Taxa de erro aparente

```
#Taxa de Erro Aparente
TEA <- 1 - (sum(diag(cvl))/sum(cvl))
TEA
```

```
## [1] 0.02
```

```
# Funcao uadratica

q <- qda(formula = Species ~ .,
         data = iris,
         prior = c(1,1,1)/3, CV=TRUE)

classificacao<- q$class
cvq <- table(iris$Species,q$class)
cvq
```

Funçaõ Quadrática

```
##
##              setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         47         3
##   virginica       0          1        49
```

```
#Taxa de Erro Aparente
TEA <- 1 - (sum(diag(cvq))/sum(cvq))
TEA
```

```
## [1] 0.02666667
```

Nesse caso, as taxas de erros ficaram bem próximas.

Aula 08: Análise Discriminante

Iris

Criando o modelo

O primeiro passo da construção da nossa árvore é separar a população entre treinamento e validação. No exemplo, separamos em 3 conjuntos de dados contendo a iris de determinada espécie; separado esses 3

conjuntos, pegamso os 5 primeiros de cada um como treinamento e o restante será para validação. Esse tipo de separação foi utilizada para que todas as esécies possuíssem a mesma proporção na construção e na validação.

```
#Particionar o conjunto de dados (Treinamento e validacao)
iris_setosa<-iris[iris$Species=="setosa",]
iris_versicolor <- iris[iris$Species=="versicolor",]
iris_virginica <- iris[iris$Species=="virginica",]
iris_train <- rbind(iris_setosa[1:25,],iris_versicolor[1:25,],
                    iris_virginica[1:25,])
iris_test <- rbind(iris_setosa[26:50,],iris_versicolor[26:50,],
                   iris_virginica[26:50,])
```

Fazendo a árvore

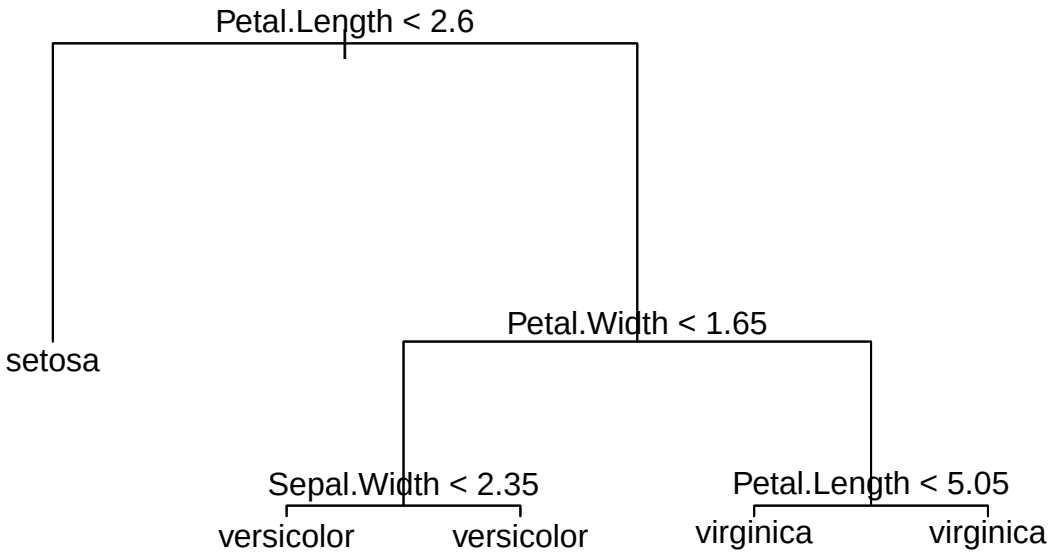
```
#Ajuste de uma arvore
arvore1 <- tree::tree(Species ~ Sepal.Width + Sepal.Length + Petal.Length +
                        Petal.Width, data = iris_train)

#Resumo (Apenas uma variael foi utilizada na construcao da arvore)
summary(arvore1)
```

```
##
## Classification tree:
## tree::tree(formula = Species ~ Sepal.Width + Sepal.Length + Petal.Length +
##     Petal.Width, data = iris_train)
## Variables actually used in tree construction:
## [1] "Petal.Length" "Petal.Width"  "Sepal.Width"
## Number of terminal nodes:  5
## Residual mean deviance:  0.143 = 10.01 / 70
## Misclassification error rate: 0.02667 = 2 / 75
```

Observamos que foi utilizado “Petal.Length”, “Petal.Width” e “Sepal.Width”, onde “Sepal.Length” acabou não sendo utilizado, como pode ser visto no gráfico:

```
#Apresentacao da arvore ajustada
plot(arvore1)
text(arvore1)
```

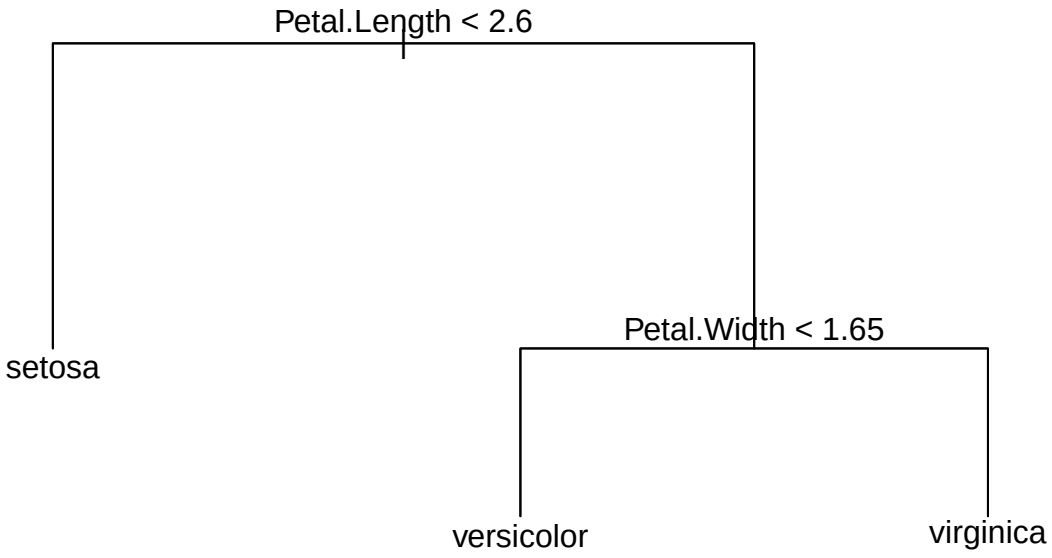


Se tivermos um comprimento de petala menor que 2.6, ele é classificado como *setosa* ($Sepal.Length < 2.6$). Observe que temos as mesmas informações em alguns nós terminais, indicando a necessidade de uma poda ($Sepal.Width < 2.35$ ou $Sepal.Width > 2.35$ classifica em *versicolor*).

```
#Verificar se e interessante realizar a poda
cv.iris=cv.tree(arvore1)
plot(cv.iris$size ,cv.iris$dev ,type="b")
```

Percebemos que a deviance (o crítério que deve ser minimizado) atinge seu mínimo em 2 nós. Mas, como temos 3 espécies, vamos considerar 3 nós.

```
#Se desejar podar
mod_poda=prune.tree(arvore1,best=3)
plot(mod_poda)
text(mod_poda, pretty =0)
```



Predição

Utilizando o modelo podado

```
#Predicao
pred_arv <- predict(mod_poda, iris_test, type="class")
head(pred_arv)
```

```
## [1] setosa setosa setosa setosa setosa setosa
## Levels: setosa versicolor virginica
```

Medidas de Performance

Para isso, vamos utilizar o pacote *caret*. Com esse pacote, fazemos a matrix de confusão (diagonal com os acertos).

```
caret::confusionMatrix(pred_arv, iris_test$Species)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   setosa versicolor virginica
##   setosa         25          0         0
```

```
##   versicolor      0         24         3
##   virginica       0          1        22
##
## Overall Statistics
##
##                Accuracy : 0.9467
##                  95% CI : (0.869, 0.9853)
##     No Information Rate : 0.3333
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.92
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: setosa Class: versicolor Class: virginica
## Sensitivity                 1.0000            0.9600           0.8800
## Specificity                 1.0000            0.9400           0.9800
## Pos Pred Value              1.0000            0.8889           0.9565
## Neg Pred Value              1.0000            0.9792           0.9423
## Prevalence                  0.3333            0.3333           0.3333
## Detection Rate              0.3333            0.3200           0.2933
## Detection Prevalence        0.3333            0.3600           0.3067
## Balanced Accuracy           1.0000            0.9500           0.9300
```

Acurácia de $0.9467 = 94.67$. O teste de hipótese *P-Value [Acc > NIR]* indica se existe um desbalanceamento entre as classes. No nosso caso, como rejeitou H_0, não existe esse desbalanceamento e o modelo discrimina bem todos os grupos. O coeficiente de *Kappa* é a acurácia ponderada por uma probabilidade de ter concordâncias aleatórias, como vários outros índices...

um dos problemas desse modelo, é o pouco poder preditivo. Para tentar aumentar esse poder, podemos utilizar o *Bagging* ou o *randomForest*.

Aumentando o Poder Preditivo (*Bagging* e *randomForest*)

```
#Ajuste bagging
bagging <- randomForest(Species~., data=iris_train,  mtry = 4)#mtry indica quantas
#variaveis serao utilizadas no processo de reamostragem
bagging
```

```
##
## Call:
##  randomForest(formula = Species ~ ., data = iris_train, mtry = 4)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 4%
## Confusion matrix:
##            setosa versicolor virginica class.error
## setosa         25          0         0        0.00
```

```
## versicolor      0         24         1        0.04
## virginica       0          2        23        0.08
```

```
#Avaliacao do modelo (Predicao)
pred_bagg <- predict(bagging, iris_test)
confusionMatrix(pred_bagg, iris_test$Species)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   setosa versicolor virginica
##   setosa         25          0         0
##   versicolor      0         23         2
##   virginica       0          2        23
##
## Overall Statistics
##
##                Accuracy : 0.9467
##                  95% CI : (0.869, 0.9853)
##     No Information Rate : 0.3333
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.92
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: setosa Class: versicolor Class: virginica
## Sensitivity                 1.0000            0.9200           0.9200
## Specificity                 1.0000            0.9600           0.9600
## Pos Pred Value              1.0000            0.9200           0.9200
## Neg Pred Value              1.0000            0.9600           0.9600
## Prevalence                  0.3333            0.3333           0.3333
## Detection Rate              0.3333            0.3067           0.3067
## Detection Prevalence        0.3333            0.3333           0.3333
## Balanced Accuracy           1.0000            0.9400           0.9400
```

```
#Random Forest
rf<- randomForest(Species~., data=iris_train,  mtry = 2)#no Random Forest, sugere-se
#utilizar um numero que seja rais de p (como temos p=4 variaveis, usaremos p=2=mtry)
rf
```

```
##
## Call:
##  randomForest(formula = Species ~ ., data = iris_train, mtry = 2)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 4%
## Confusion matrix:
##            setosa versicolor virginica class.error
```

```
## setosa         25          0         0        0.00
## versicolor      0         24         1        0.04
## virginica       0          2        23        0.08
```

```
#Avaliacao do modelo (Predicao)
pred_rf <- predict(rf, iris_test)
confusionMatrix(pred_rf, iris_test$Species)
```

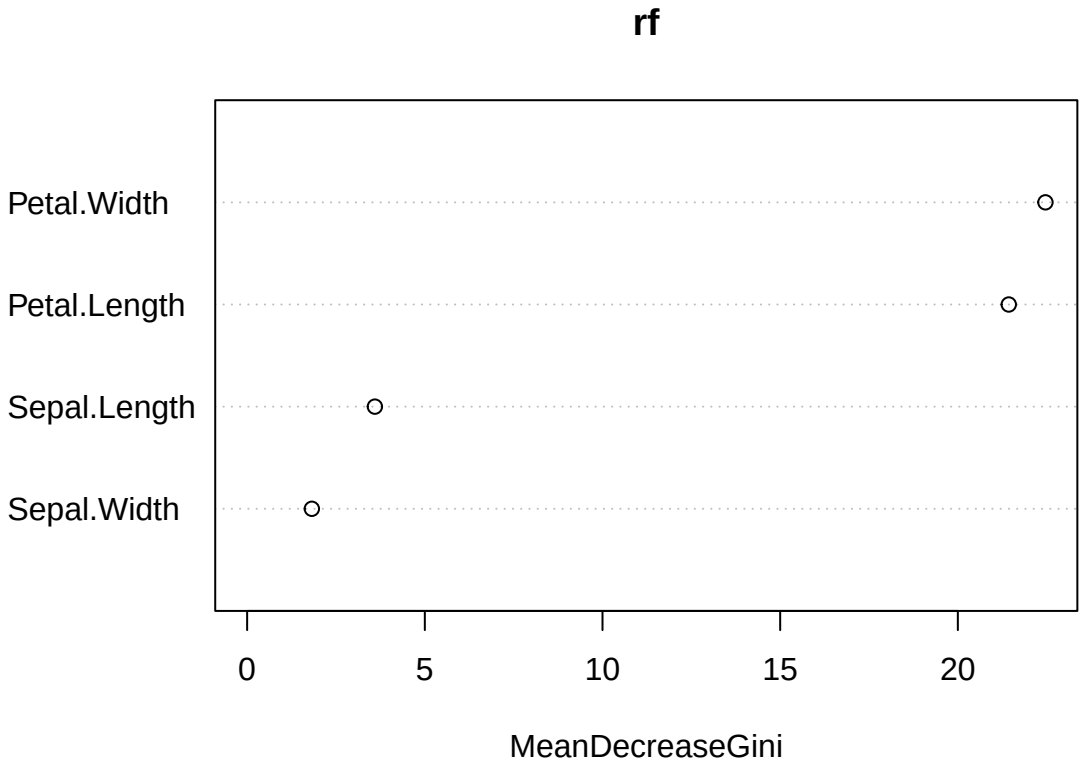
```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   setosa versicolor virginica
##   setosa         25          0         0
##   versicolor      0         23         2
##   virginica       0          2        23
##
## Overall Statistics
##
##                Accuracy : 0.9467
##                  95% CI : (0.869, 0.9853)
##     No Information Rate : 0.3333
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.92
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: setosa Class: versicolor Class: virginica
## Sensitivity                 1.0000            0.9200           0.9200
## Specificity                 1.0000            0.9600           0.9600
## Pos Pred Value              1.0000            0.9200           0.9200
## Neg Pred Value              1.0000            0.9600           0.9600
## Prevalence                  0.3333            0.3333           0.3333
## Detection Rate              0.3333            0.3067           0.3067
## Detection Prevalence        0.3333            0.3333           0.3333
## Balanced Accuracy           1.0000            0.9400           0.9400
```

```
#Importancia (Baseado nas amostras out-of_bag)
i_mod_rf <-importance(rf)
i_mod_rf
```

```
##              MeanDecreaseGini
## Sepal.Length         3.594658
## Sepal.Width          1.820406
## Petal.Length        21.433438
## Petal.Width         22.464605
```

Verificando a importância das variáveis no modelo de classfiicação. (maiores valores indicam as variáveis mais importantes).

```
varImpPlot (rf)
```



Avaliação referente ao conteúdo ministrado na Semana 4

Questão 1

Correto

Atingiu 1,00 de 1,00

Marcar questão

São medidas que podem ser utilizadas na construção de uma árvores de decisão

- [x] a. Taxa de Erro Aparente ✔
- [x] b. Deviance ✔
- [ ] c. Nível de significância
- [ ] d. Taxa de Acerto
- [x] e. Índice de Gini ✔

Questão 2

Correto

Atingiu 1,00 de 1,00

Marcar questão

Para ajuste das funções discriminantes é necessário que a função densidade de probabilidade sejam conhecidas.

Escolha uma opção:

- (•) Verdadeiro ✔
- ( ) Falso

Questão 3

Correto

Atingiu 1,00 de 1,00

Marcar questão

A derivação das funções discriminantes se baseiam na minimização do custo médio de classificação incorreta.

Escolha uma opção:

- Verdadeiro ✔
- Falso

Questão 4

Correto

Atingiu 1,00 de 1,00

Marcar questão

Considerando duas populações normais com matrizes de covariâncias homogêneas devemos classificar um indivíduo x na população 1 se:

$$-\frac{1}{2}\mathbf{x}^T(\boldsymbol{\Sigma}_1^{-1}-\boldsymbol{\Sigma}_2^{-1})\mathbf{x}+(\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}_1^{-1}-\boldsymbol{\mu}_2^T\boldsymbol{\Sigma}_2^{-1})\mathbf{x}-\delta \geq \ln\left\{\left[\frac{C(1|2)}{C(2|1)}\right]\left(\frac{p_2}{p_1}\right)\right\},$$

Escolha uma opção:

- Verdadeiro
- Falso ✔

Questão 5

Correto

Atingiu 1,00 de 1,00

Marcar questão

As funções discriminantes quadráticas não são sensíveis a falta de normalidade.

Escolha uma opção:

- Verdadeiro
- Falso ✔