Curso: Data Science & Business Intelligence
Unidade Curricular: Data Mining
Docente: Roberto Vita

TÓPICO 2

Modelos de Classificação

Wine Dataset (*UCI*)

- KNN
- Naive Bayes
- Árvores de Decisão

Trabalho realizado por:

Marta Lobo

Marta Barros

KNN

KNN é um método de classificação supervisionado, não paramétrico e não linear. Este classifica uma nova observação a partir de 'k' vizinhos mais próximos. A distância entre os vizinhos mais próximos neste trabalho é calculada a partir da distância euclidiana, no entanto existem vários métodos.

O dataset que utilizamos foi o Wine Data Set disponível no repositório UCI de Machine Learning. O nosso dataset tem treze variáveis input, todas elas numéricas (**Figura 1**): Alcohol, Malic Acid, Ash, Alcalinity of Ash, Magnesium, Total Phenols, Flavanoids, Nonflavanoids Phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of Diluted Wines e Proline.

```
> dataset<-wine
> str(dataset)
tibble [177 x 14] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ Cultivars           : num [1:177] 1 1 1 1 1 1 1 1 1 1 ...
 $ Alcohol             : num [1:177] 13.2 13.2 14.4 13.2 14.2 ...
 $ Malic_acid          : num [1:177] 1.78 2.36 1.95 2.59 1.76 1.87 2.15 1.64 1.35 2.16 ...
 $ Ash                 : num [1:177] 2.14 2.67 2.5 2.87 2.45 2.45 2.61 2.17 2.27 2.3 ...
 $ Alcalinity_of_ash   : num [1:177] 11.2 18.6 16.8 21 15.2 14.6 17.6 14 16 18 ...
 $ Magnesium           : num [1:177] 100 101 113 118 112 96 121 97 98 105 ...
 $ Total_phenols       : num [1:177] 2.65 2.8 3.85 2.8 3.27 2.5 2.6 2.8 2.98 2.95 ...
 $ Flavanoids          : num [1:177] 2.76 3.24 3.49 2.69 3.39 2.52 2.51 2.98 3.15 3.32 ...
 $ Nonflavanoid phenols: num [1:177] 0.26 0.3 0.24 0.39 0.34 0.3 0.31 0.29 0.22 0.22 ...
 $ Proanthocyanins     : num [1:177] 1.28 2.81 2.18 1.82 1.97 1.98 1.25 1.98 1.85 2.38 ...
 $ Color_intensity     : num [1:177] 4.38 5.68 7.8 4.32 6.75 5.25 5.05 5.2 7.22 5.75 ...
 $ Hue                 : num [1:177] 1.05 1.03 0.86 1.04 1.05 1.02 1.06 1.08 1.01 1.25 ...
 $ diluted_wines       : num [1:177] 3.4 3.17 3.45 2.93 2.85 3.58 3.58 2.85 3.55 3.17 ...
 $ Proline             : num [1:177] 1050 1185 1480 735 1450 ...
```

Figura 1. Descrição do tipo de variáveis (num) do dataset wine, distribuído em 177 linhas e 14 colunas [177x 14].

Posteriormente, transformou-se a variável *Cultivars* em factor para utilização em KNN (**Figura 2**). Este método implica o cálculo de distâncias entre pontos, pelo que se devem utilizar apenas variáveis numéricas como preditoras. No entanto, a variável de outcome ou classificadora deve permanecer como factor.

```
> #Transformar variável Cultivars em factor
> levels(dataset$Cultivars)
NULL
> dataset$Cultivars<-factor(dataset$Cultivars)
> levels(dataset$Cultivars)
[1] "1" "2" "3"
```

Figura 2. Fatorização e atribuição de níveis na variável classificadora.

Foi feita uma breve análise estatística ao dataset original, onde se estabeleceu o mínimo, o máximo, a média, a mediana, o primeiro e terceiro quartil para cada uma das variáveis (**Figura 3**).

```
> summary(wine)
   Cultivars         Alcohol        Malic_acid         Ash        Alcalinity_of_ash
 Min.   :1.000   Min.   :11.03   Min.   :0.74   Min.   :1.360   Min.   :10.60
 1st Qu.:1.000   1st Qu.:12.36   1st Qu.:1.60   1st Qu.:2.210   1st Qu.:17.20
 Median :2.000   Median :13.05   Median :1.87   Median :2.360   Median :19.50
 Mean   :1.944   Mean   :12.99   Mean   :2.34   Mean   :2.366   Mean   :19.52
 3rd Qu.:3.000   3rd Qu.:13.67   3rd Qu.:3.10   3rd Qu.:2.560   3rd Qu.:21.50
 Max.   :3.000   Max.   :14.83   Max.   :5.80   Max.   :3.230   Max.   :30.00
   Magnesium       Total_phenols     Flavanoids    Nonflavanoid phenols Proanthocyanins
 Min.   : 70.00   Min.   :0.980   Min.   :0.340   Min.   :0.1300       Min.   :0.410
 1st Qu.: 88.00   1st Qu.:1.740   1st Qu.:1.200   1st Qu.:0.2700       1st Qu.:1.250
 Median : 98.00   Median :2.350   Median :2.130   Median :0.3400       Median :1.550
 Mean   : 99.59   Mean   :2.292   Mean   :2.023   Mean   :0.3623       Mean   :1.587
 3rd Qu.:107.00   3rd Qu.:2.800   3rd Qu.:2.860   3rd Qu.:0.4400       3rd Qu.:1.950
 Max.   :162.00   Max.   :3.880   Max.   :5.080   Max.   :0.6600       Max.   :3.580
 Color_intensity       Hue         diluted_wines      Proline
 Min.   : 1.280   Min.   :0.480   Min.   :1.270   Min.   : 278.0
 1st Qu.: 3.210   1st Qu.:0.780   1st Qu.:1.930   1st Qu.: 500.0
 Median : 4.680   Median :0.960   Median :2.780   Median : 672.0
 Mean   : 5.055   Mean   :0.957   Mean   :2.604   Mean   : 745.1
 3rd Qu.: 6.200   3rd Qu.:1.120   3rd Qu.:3.170   3rd Qu.: 985.0
 Max.   :13.000   Max.   :1.710   Max.   :4.000   Max.   :1680.0
```

Figura 3. Estatística sumarizada do dataset original.

Para colocar os dados na mesma escala e tornar possível o cálculo de distâncias implicado no método KNN, normalizamos o dataset (**Figura 4**).

```
> #normalizar dados
> normalize <- function(x) {
+   + return((x-min(x))/(max(x)-min(x)))}
> dataset_x_norm <- as.data.frame(lapply(dataset[,2:14], normalize))
> #Juntar coluna Cultivars aos dados normalizados
> dataset_norm <- cbind(dataset_x_norm, dataset$Cultivars)
> summary(dataset_norm)
    Alcohol         Malic_acid          Ash         Alcalinity_of_ash   Magnesium
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000    Min.   :0.0000
 1st Qu.:0.3500   1st Qu.:0.1700   1st Qu.:0.4545   1st Qu.:0.3402    1st Qu.:0.1957
 Median :0.5316   Median :0.2233   Median :0.5348   Median :0.4588    Median :0.3043
 Mean   :0.5168   Mean   :0.3162   Mean   :0.5381   Mean   :0.4596    Mean   :0.3216
 3rd Qu.:0.6947   3rd Qu.:0.4664   3rd Qu.:0.6417   3rd Qu.:0.5619    3rd Qu.:0.4022
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000    Max.   :1.0000
 Total_phenols      Flavanoids     Nonflavanoid.phenols Proanthocyanins  Color_intensity
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000       Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.2621   1st Qu.:0.1814   1st Qu.:0.2642       1st Qu.:0.2650   1st Qu.:0.1647
 Median :0.4724   Median :0.3776   Median :0.3962       Median :0.3596   Median :0.2901
 Mean   :0.4525   Mean   :0.3552   Mean   :0.4383       Mean   :0.3713   Mean   :0.3221
 3rd Qu.:0.6276   3rd Qu.:0.5316   3rd Qu.:0.5849       3rd Qu.:0.4858   3rd Qu.:0.4198
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000       Max.   :1.0000   Max.   :1.0000
      Hue         diluted_wines       Proline       dataset$Cultivars
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   1:58
 1st Qu.:0.2439   1st Qu.:0.2418   1st Qu.:0.1583   2:71
 Median :0.3902   Median :0.5531   Median :0.2810   3:48
 Mean   :0.3878   Mean   :0.4888   Mean   :0.3332
 3rd Qu.:0.5203   3rd Qu.:0.6960   3rd Qu.:0.5043
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
```

Figura 4. Estatística associada ao dataset normalizado, com junção da coluna original fatorizada dos cultivars (*cbind*).

De seguida, definimos as etiquetas de classificação (*Wine_Labels*) e o modelo de KNN, com k=13, com base na raiz quadrada do número de linhas do nosso dataset. Optámos também por um número ímpar, para evitar empates no critério de voto para o 'vizinho' mais próximo (**Figura 5**).

```
> library(class)
> library("caret")
> wineTrain <- training_set[,-1]
> wineTest <- test_set[,-1]
> wineTrain_label <- training_set[,1 , drop = TRUE]
> wineTest_label <- test_set[,1 , drop = TRUE]
> set.seed(123)
> wineknns<- knn(train = wineTrain, test=wineTest, cl=wineTrain_label, k=13,prob = TRUE)
> wineknns
 [1] 3 1 1 1 1 3 3 1 1 1 1 2 3 2 3 3 3 3 3 2 2 2 2 2 3 2 3 1 2 3
attr(,"prob")
 [1] 0.6153846 1.0000000 1.0000000 1.0000000 1.0000000 0.5384615 0.4615385 0.4615385
 [9] 1.0000000 0.9230769 1.0000000 0.5384615 0.6923077 0.5384615 0.5384615 0.6923077
[17] 0.6923077 0.6153846 0.6923077 0.5384615 0.7692308 1.0000000 0.9230769 0.9230769
[25] 1.0000000 0.6153846 0.5384615 0.5384615 0.4615385 0.7692308 0.6153846
Levels: 1 2 3
```

Figura 5. Definição do modelo **`wineknns`** `<- knn (train = wineTrain, test=wineTest, cl=wineTrain_label, k=13, prob = TRUE).`

Tendo em vista o cálculo da acurácia e respetiva avaliação do modelo, gerou-se a matriz de confusão entre os valores do **`wineknns`** e os valores dos *Cultivars* (1, 2, 3) da base de dados de teste (**Figura 6**).

```
> #matriz de confusão
> table(wineknns, test_set$Cultivars)

wineknns 1 2 3
       1 8 0 1
       2 0 8 2
       3 3 6 3
```

Figura 6. Matriz de confusão para o método KNN.

Tentou-se ainda fazer o *cross-validation* do valor final da acurácia, com um fold 10, isto é, testar 10 vezes a nossa base de dados de treino versus a de teste, verificando se existe ou não flutuação da eficácia do modelo (**Figura 7**). O valor final de acurácia para um k=13, exemplo testado, foi de 61% (**Figura 8**).

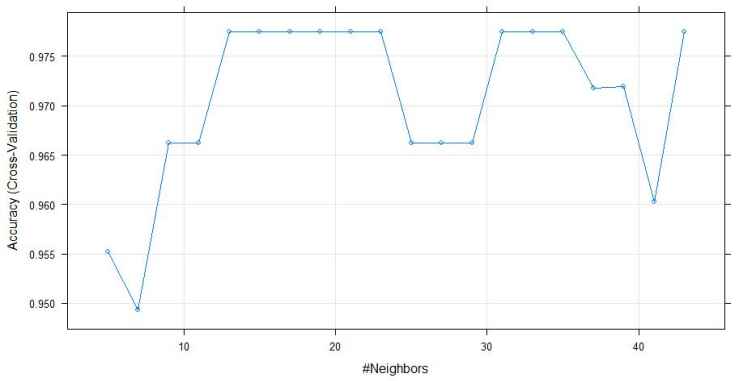

Figura 7. Cross-Validation da acurácia, com um fold de 10.

```
> model$bestTune
    k
20 43
> acuracia <- 100 * sum(wineTest_label  == wineknns)/NROW(wineTest_label)
> acuracia
[1] 61.29032
```

Figura 8. Ajuste do modelo aos k (20-43) e cálculo da acurácia para k=13 (61%).

NAIVE BAYES

O Naive Bayes é um algoritmo de aprendizagem supervisionada, cujo foco assenta sobre a probabilidade condicional e a função densidade de probabilidade multivariada. É também um modelo que aceita relações não lineares entre variáveis, pelo que é bastante flexível.

Este parte da assumpção que as variáveis são independentes e distribuídas normalmente caso sejam numéricas. Caso a normalidade seja violada, a discretização pode suprir o problema, sem perda de informação relativamente às características, pelo que, optamos por utilizar o cultivar em três classes.

Na nossa avaliação, serão utilizadas as bibliotecas *GGally*, *KlaR*, *caret* e *e1071*.

Efetuou-se a análise exploratória dos dados através da função *ggpairs*(), obtendo-se um gráfico de relação entre as variáveis do dataset (**Figura 9**). Uma vez que as variáveis em questão são numéricas, a diagonal do nosso gráfico estabelece gráficos de densidade para cada uma delas. Abaixo da linha diagonal, temos uma série de gráficos de dispersão e acima da mesma

encontram-se definidos os coeficientes de correlação entre as variáveis que, na grande maioria dos casos, são significativos.

Relativamente à primeira coluna e considerando **Cultivars** como elemento de classificação, percebemos através dos gráficos de dispersão que apenas se distribui em 3 zonas, correspondentes aos três tipos de cultivares existentes.

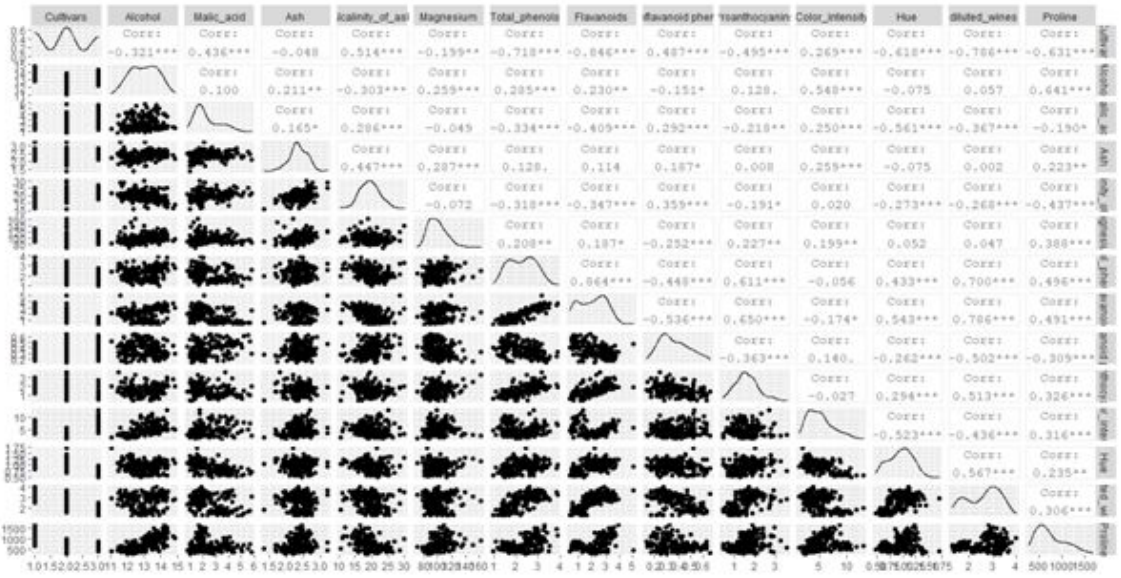

Figura 9. Matriz de gráficos associada ao dataset geral *ggpairs* (winebay).

De seguida, calcularam-se as probabilidades condicionais *a priori* para cada uma das variáveis com base no modelo Naive Bayes e o classificador discreto *Cultivars*. Obteve-se uma distribuição de 32%, 39% e 29% (**Figura 10**) para os cultivares 1, 2 e 3, respetivamente.

```
> model<-NaiveBayes(Cultivars~.,data=training_set, fL = 1, prob=TRUE)
> model
$apriori
grouping
        1         2         3
0.3219178 0.3904110 0.2876712

$tables
$tables$Alcohol
      [,1]      [,2]
1 13.72809 0.4351893
2 12.25772 0.5610609
3 13.17952 0.5441258

$tables$Malic_acid
      [,1]      [,2]
1 2.022766 0.7063922
2 1.998421 1.0803898
3 3.270238 1.0320675

$tables$Ash
      [,1]      [,2]
1 2.460000 0.2268115
2 2.254035 0.3145454
3 2.434524 0.1878012

$tables$Alcalinity_of_ash
      [,1]     [,2]
1 17.00638 2.548350
2 20.30526 3.269252
3 21.22619 2.266410

$tables$Magnesium
       [,1]     [,2]
1 105.93617 10.07020
2  96.66667 17.53296
3  99.11905 11.44489

$tables$Total_phenols
      [,1]      [,2]
1 2.871489 0.3463661
2 2.290526 0.5732128
3 1.682381 0.3566088

$tables$Flavanoids
      [,1]      [,2]
1 2.9985106 0.4131374
2 2.1343860 0.7173239
3 0.7785714 0.2909075

$tables$Nonflavanoid.phenols
      [,1]       [,2]
1 0.2902128 0.07057187
2 0.3573684 0.13155018
3 0.4490476 0.12137042

$tables$Proanthocyanins
      [,1]      [,2]
1 1.870000 0.4271493
2 1.721404 0.5975499
3 1.166905 0.4300113

$tables$Color_intensity
      [,1]      [,2]
1 5.562766 1.2552909
2 3.082982 0.9393785
3 7.545238 2.2744908

$tables$Hue
      [,1]      [,2]
1 1.0538298 0.1262426
2 1.0709825 0.2076932
3 0.6828571 0.1108126

$tables$diluted_wines
      [,1]      [,2]
1 3.161277 0.3331119
2 2.788947 0.4819999
3 1.683810 0.2753707

$tables$Proline
       [,1]     [,2]
1 1126.0000 210.6415
2  516.7719 162.7081
3  631.4286 113.5045
```

Figura 10. Probabilidade condicionada *a priori.*

No sentido de avaliar a adequação do modelo, gerou-se uma matriz de confusão entre os Y esperados e a variável **Cultivars** associada à base de treino, Y reais (**Figura 11**). Concluiu-se uma acurácia de 99%, bastante superior aos outros modelos de classificação (**Figura 8 e 15**). Quanto maior a independência entre as variáveis, maior a eficácia do modelo Naive Bayes.

```
> confusionMatrix(y_pred, training_set$Cultivars)
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3
         1 47  0  0
         2  0 56  0
         3  0  1 42

Overall Statistics

               Accuracy : 0.9932
                 95% CI : (0.9624, 0.9998)
    No Information Rate : 0.3904
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9897

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3
Sensitivity            1.0000   0.9825   1.0000
Specificity            1.0000   1.0000   0.9904
Pos Pred Value         1.0000   1.0000   0.9767
Neg Pred Value         1.0000   0.9889   1.0000
Prevalence             0.3219   0.3904   0.2877
Detection Rate         0.3219   0.3836   0.2877
Detection Prevalence   0.3219   0.3836   0.2945
Balanced Accuracy      1.0000   0.9912   0.9952
```

Figura 11. Matriz de confusão, estatística geral e estatística por classe.

ÁRVORES DE DECISÃO

O dataset foi dividido em treino e teste à semelhança dos métodos anteriores, obtendo-se a divisão exibida na **Figura 12**.

```
> #divisão do dataset
> set.seed(123)
> split <- sample(2, nrow(winetree), replace = TRUE, prob = c(0.8, 0.2))
> training_set <- winetree[split == 1, ]
> test_set <- winetree[split == 2, ]
> nrow(training_set)
[1] 146
> nrow(test_set)
[1] 31
```

Figura 12. Divisão da base de dados em treino (146) e teste (31).

Depois inicializou-se a biblioteca *rpart* e definiu-se o modelo de árvore de decisão para o nosso training set (**Figura 13**).

```
> library(rpart)
> arvore <- rpart(Cultivars ~., training_set )
> arvore
n= 146

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 146 89 2 (0.32191781 0.39041096 0.28767123)
   2) Color_intensity< 3.825 52  3 2 (0.05769231 0.94230769 0.00000000) *
   3) Color_intensity>=3.825 94 50 1 (0.46808511 0.08510638 0.44680851)
     6) Flavanoids>=1.58 51  7 1 (0.86274510 0.13725490 0.00000000)
      12) Proline>=737 43  0 1 (1.00000000 0.00000000 0.00000000) *
      13) Proline< 737 8  1 2 (0.12500000 0.87500000 0.00000000) *
     7) Flavanoids< 1.58 43  1 3 (0.00000000 0.02325581 0.97674419) *
```

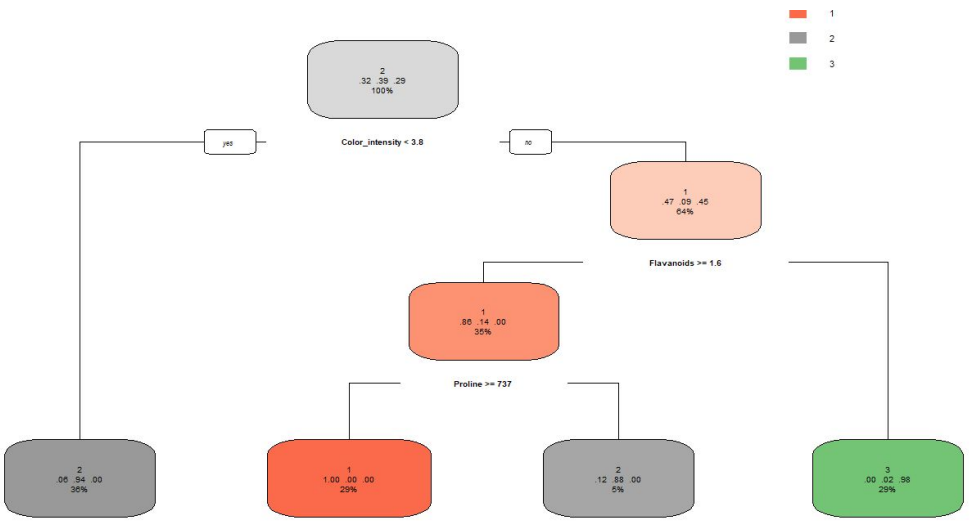

Figura 13. Divisão da árvore de decisão em raiz e respetivas ramificações.

```
> library(caret)
> confusionMatrix(y_pred, test_set$Cultivars)
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3
         1  9  0  0
         2  2 14  0
         3  0  0  6

Overall Statistics

               Accuracy : 0.9355
                 95% CI : (0.7858, 0.9921)
    No Information Rate : 0.4516
    P-Value [Acc > NIR] : 1.438e-08

                  Kappa : 0.897

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3
Sensitivity            0.8182   1.0000   1.0000
Specificity            1.0000   0.8824   1.0000
Pos Pred Value         1.0000   0.8750   1.0000
Neg Pred Value         0.9091   1.0000   1.0000
Prevalence             0.3548   0.4516   0.1935
Detection Rate         0.2903   0.4516   0.1935
Detection Prevalence   0.2903   0.5161   0.1935
Balanced Accuracy      0.9091   0.9412   1.0000
```

Figura 14. Matriz de confusão, estatística geral e estatística por classe.

Seguindo as mesmas etapas, testaram-se dois outros métodos de definição de modelos de árvore de decisão, nomeadamente *prunning* e *bagging* - ver **Anexo**. O método mais eficaz parece ser o *rpart*, uma vez que o *bagging* testa várias vezes os mesmos valores, daí a acurácia igual a 1 não ser indicativa de melhor performance (**Figura 15**).

	Método	Código	Acurácia
Árvores de decisão	rpart	confusionMatrix(y_pred, test_set$Cultivars)	0.9355
	prunning	confusionMatrix(y_pred_prune, test_set$Cultivars)	0.871
	bagging	confusionMatrix(y_pred_bag, test_set$Cultivars)	1

Figura 15. Tabela-resumo dos vários métodos aplicados ao modelo de árvore de decisão e acurácia correspondente.

ANEXO

1. Prunning

```
> #prunning
>
> nova_arvore <- prune(arvore, cp=0.1)
> nova_arvore
n= 146

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 146 89 2 (0.32191781 0.39041096 0.28767123)
  2) Color_intensity< 3.825 52  3 2 (0.05769231 0.94230769 0.00000000) *
  3) Color_intensity>=3.825 94 50 1 (0.46808511 0.08510638 0.44680851)
    6) Flavanoids>=1.58 51  7 1 (0.86274510 0.13725490 0.00000000) *
    7) Flavanoids< 1.58 43  1 3 (0.00000000 0.02325581 0.97674419) *
```

```
> y_pred_prune <- predict(nova_arvore, newdata = test_set, type ="class")
> y_pred_prune <- factor(y_pred_prune , levels = c("1", "2","3"))
> confusionMatrix(y_pred_prune, test_set$Cultivars)
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3
         1 10  3  0
         2  1 11  0
         3  0  0  6

Overall Statistics

               Accuracy : 0.871
                 95% CI : (0.7017, 0.9637)
    No Information Rate : 0.4516
    P-Value [Acc > NIR] : 1.532e-06

                  Kappa : 0.798

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3
Sensitivity            0.9091   0.7857   1.0000
Specificity            0.8500   0.9412   1.0000
Pos Pred Value         0.7692   0.9167   1.0000
Neg Pred Value         0.9444   0.8421   1.0000
Prevalence             0.3548   0.4516   0.1935
Detection Rate         0.3226   0.3548   0.1935
Detection Prevalence   0.4194   0.3871   0.1935
Balanced Accuracy      0.8795   0.8634   1.0000
```

2. Bagging

```
> library(ipred)
> set.seed(123)
> bag_arvore <- bagging(Cultivars ~., training_set, nbagg = 100)
> y_pred_bag <- predict(bag_arvore, newdata = test_set, type ="class")
> y_pred_bag <- factor(y_pred_bag , levels = c("1", "2","3"))
> confusionMatrix(y_pred_bag, test_set$Cultivars)
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3
         1 11  0  0
         2  0 14  0
         3  0  0  6

Overall Statistics

               Accuracy : 1
                 95% CI : (0.8878, 1)
    No Information Rate : 0.4516
    P-Value [Acc > NIR] : 1.985e-11

                  Kappa : 1

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3
Sensitivity            1.0000   1.0000   1.0000
Specificity            1.0000   1.0000   1.0000
Pos Pred Value         1.0000   1.0000   1.0000
Neg Pred Value         1.0000   1.0000   1.0000
Prevalence             0.3548   0.4516   0.1935
Detection Rate         0.3548   0.4516   0.1935
Detection Prevalence   0.3548   0.4516   0.1935
Balanced Accuracy      1.0000   1.0000   1.0000
```