Determinação de Classes e Critérios de Classificaçaõ de Solo - Profundidade B

Joel Maurício Corrêa da Rosa

2 de fevereiro de 2007

Neste relatório é apresentada uma seqüência de operações no software R feita com o objetivo de primeiramente criar uma classificação para diferentes observações de solo e apresentar estas regras de classificação através de uma árvore de decisão binária.

1 Métodos de Clusterização

Algoritmos de clusterização, ou agrupamento, são métodos utilizados para dividir um conjunto de n objetos em k grupos que sejam internamente homogêneos e heterogêneos entre si em relação à um conjunto de variáveis. Estes algoritmos são divididos em duas classes: hierárquicos e não-hierárquicos.

Neste trabalho são aplicadas duas técnicas de clusterização; uma do tipo *crisp* na qual o conjunto de dados é particionado em k subconjuntos e cada objeto deve pertencer somente a um dos subconjuntos e outra técnica, classificada como *fuzzy*,em que cada objeto pertence a todos os k grupos, mas com diferentes graus de pertinência.

Nos algoritmos do tipo *fuzzy*, os objetos pertencem a todos os grupos, mas com diferentes graus de pertinência. Desta forma, ao objeto são associados o conjunto e o seu grau de pertinência àquele conjunto.

Os agrupamentos são formados mediante a avaliação de uma medida de distância ou dissimilaridade.

2 Agrupamentos no Solo Segundo Atributos Químicos

A análise a ser apresentada nesta seção refere-se à formação de agrupamentos no solo com base em vetores observados do solo para 9 variáveis 'químicas'.

Os comandos abaixo, procedem a leitura dos dados e suas respectivas coordenadas para o processamento da análise de agrupamentos.

```
> rm(list = ls())
> dados.ori <- read.table("http://leg.ufpr.br/~joel/pesquisa/dados/Quimicos_C2.txt",
+     h = T, sep = "", dec = ",")
> names(dados.ori)
```

```
 [1] "X"      "Y"      "Al"     "Ca"     "Mg"     "PhH2O"  "PhKCL"  "K"
 [9] "P"      "C"      "AB"     "H"      "Prof"   "Regiao"
```

```
> head(dados.ori)
```

```
       X       Y  Al   Ca   Mg PhH2O PhKCL    K   P    C AB   H Prof Regiao
1 688713 7190023 2.0 1.55 1.51  5.15  4.20 0.06 0.4 12.4  2 3.9   22      1
2 688694 7190000 3.5 1.05 1.06  5.10  4.30 0.10 0.2 16.0  2 6.2   35      1
3 688457 7189578 1.4 0.85 0.41  5.30  4.25 0.04 0.2  8.8  2 2.7   38      3
4 688790 7190115 6.0 0.23 0.03  4.85  4.05 0.08 0.4  6.9  2 2.0   40      1
5 688592 7189756 3.0 0.61 0.62  5.10  4.25 0.04 0.6 12.4  2 4.7   42      3
6 688601 7189843 2.3 1.13 0.98  5.25  4.20 0.03 0.2 10.6  2 4.6   45      2
```

Para evitar problemas em função das diferentes escalas de medição das variáveis químicas, abaixo é feita a padronização das variáveis, subtraindo-as da média observada e dividindo pelos respectivos desvios-padrão.

Ressalta-se aqui que alguns pacotes do R para clusterização contém em suas funções, parâmetros para a padronização das variáveis.

```
> DadosClust <- dados.ori[, c(3:10, 12)]
> medias <- apply(DadosClust, 2, mean)
> desvpad <- apply(DadosClust, 2, sd)
> DadosClust.p <- DadosClust
> for (i in 1:9) {
+     DadosClust.p[i] <- DadosClust.p[i] - medias[i]
+ }
> for (i in 1:9) {
+     DadosClust.p[i] <- DadosClust.p[i]/desvpad[i]
+ }
```

Conforme pode ser verificado, através da matriz de correlações, há indícios de fortes associações entre variáveis químicas tal como ocorre entre Alumínio (Al) e Potássio (K) cujo coeficiente de correlação linear é igual a -0,7464

```
> cor(DadosClust)
```

	Al	Ca	Mg	PhH2O	PhKCL	K
Al	1.00000000	-0.145865155	-0.16177153	-0.41532358	-0.79025474	0.160913458
Ca	-0.14586515	1.000000000	0.50256315	0.12357576	0.14640758	-0.004615275
Mg	-0.16177153	0.502563149	1.00000000	0.12857666	0.21424913	0.018640192
PhH2O	-0.41532358	0.123575762	0.12857666	1.00000000	0.41179120	0.049801649
PhKCL	-0.79025474	0.146407580	0.21424913	0.41179120	1.00000000	0.022025355
K	0.16091346	-0.004615275	0.01864019	0.04980165	0.02202536	1.000000000
P	0.11018646	0.086904836	0.12811349	-0.09723479	-0.14291278	0.129478407
C	-0.02592455	0.045412441	0.04489215	0.13487250	0.02944880	-0.001993542
H	-0.23320377	0.146490918	0.12154734	0.28470817	0.22827132	0.011006393

	P	C	H
Al	0.110186462	-0.025924553	-0.23320377
Ca	0.086904836	0.045412441	0.14649092
Mg	0.128113493	0.044892149	0.12154734
PhH2O	-0.097234791	0.134872497	0.28470817
PhKCL	-0.142912785	0.029448802	0.22827132
K	0.129478407	-0.001993542	0.01100639
P	1.000000000	0.004206076	0.01568585
C	0.004206076	1.000000000	0.45237489
H	0.015685850	0.452374893	1.00000000

Abaixo, são carregados 3 pacotes no R para utilização de algoritmos de clusterização, definido o número de clusters e também as cores que representarão os clusters na visualização espacial.

```
> require(flexclust)

[1] TRUE

> require(cclust)

[1] TRUE

> require(cluster)

[1] TRUE

> numclust = 4
> ind.c <- colors()[c(562, 45, 499, 10, 614)]
> ind.c

[1] "royalblue"    "cadetblue3"   "orange1"      "aquamarine2"  "springgreen4"
```

2.1 Clusterização Fuzzy Através da Função fanny

Utiliza-se abaixo a função fanny presente no pacote cluster para a determinação dos clusters nebulosos.

Em função da expectativa, anterior a coleta de dados, em relação à quantidade de diferentes classes de solo na região da coleta de dados, foi feita a opção inicial por K=4 agrupamentos.

Arbitrariamente foi definido o parâmetro memb.exp que determina o grau de nebulosidade da clusterização.

```
> clust.info.fuz <- fanny(DadosClust.p, numclust, memb.exp = 1.3,
+     maxit = 3000, tol = 1e-20)
> clust.fuz <- clust.info.fuz$clustering
> max.memb <- apply(clust.info.fuz$membership, 1, max)
```

A partir dos graus de pertinência, é possível formar agrupamentos, alocando as observações ao grupo ao qual está associado o seu maior grau de pertinência. Veja, por exemplo, os grau de pertinência da observação 48 aos K=4 diferentes clusters.

```
> memb.48 <- clust.info.fuz$membership[48, ]
> memb.48

[1] 0.16088709 0.65617286 0.04778152 0.13515853

> clust.48 <- which(memb.48 == max(memb.48))
> clust.48

[1] 2
```

Conforme a seqüência de pertinências mostrada acima, a observação 48 deve ser alocada ao cluster 2 que detém a maior pertinência (0.6562). De acordo com este procedimento de alocação das observações aos grupos, abaixo mostra-se a distribuição de freqüências dos agrupamentos.

```
> table(clust.info.fuz$clustering)

 1  2  3  4
36 39 33 25

> DadosClust.p$fuzclust <- factor(clust.info.fuz$clustering)
> head(DadosClust.p)
          Al         Ca          Mg      PhH2O       PhKCL          K
1 -0.3163010  1.1382793  2.86548522 -0.1963890 -0.59685159  0.1230950
2  0.6933319  0.2265041  1.72610647 -0.3649034 -0.03782862  0.8349054
3 -0.7201542 -0.1382059  0.08033715  0.3091543 -0.31734010 -0.2328102
4  2.3760533 -1.2688071 -0.88180491 -1.2074754 -1.43538603  0.4790002
5  0.3567876 -0.5758580  0.61204724 -0.3649034 -0.31734010 -0.2328102
6 -0.1143745  0.3723882  1.52355024  0.1406398 -0.59685159 -0.4107627
          P           C           H fuzclust
1  0.1899219  0.24588662 -0.02551318        1
2 -0.4187435  0.70220643  1.21329361        1
3 -0.4187435 -0.21043320 -0.67184716        2
4  0.1899219 -0.45126865 -1.04887531        3
5  0.7985873  0.24588662  0.40537614        4
6 -0.4187435  0.01772671  0.35151497        4

> means <- by(DadosClust.p[, -10], DadosClust.p$fuzclust, mean)
> centroid <- NULL
> for (i in 1:numclust) {
+     centroid <- rbind(centroid, means[[i]])
+ }
```

2.2 Clusterização Crisp através do Método das K-Médias

Utiliza-se abaixo o comando `kcca` para a determinação dos clusters crisps. O algoritimo das k-médias é um procedimento iterativo que necessita a definição de valores iniciais para os centróides de cada cluster. A escolha por estes valores iniciais foi feita com base nos resultados obtidos através da fuzzy-clusterização. Deste modo, os centróides obtidos ao final do processo anterior serão utilizados como valores iniciais neste algoritmo.

```
> require(cclust)

[1] TRUE

> clust.info.crisp <- kcca(DadosClust.p[, -10], centroid)
> clust.info.crisp

kcca object of family 'kmeans'

call:
kcca(x = DadosClust.p[, -10], k = centroid)

cluster sizes:

 1  2  3  4
21 48 33 31

> clust.crisp <- predict(clust.info.crisp)
```

Para entender melhor a característica de cada cluster, na sequência é apresentado, através de um gráfico de barras, os centróides de cada cluster.

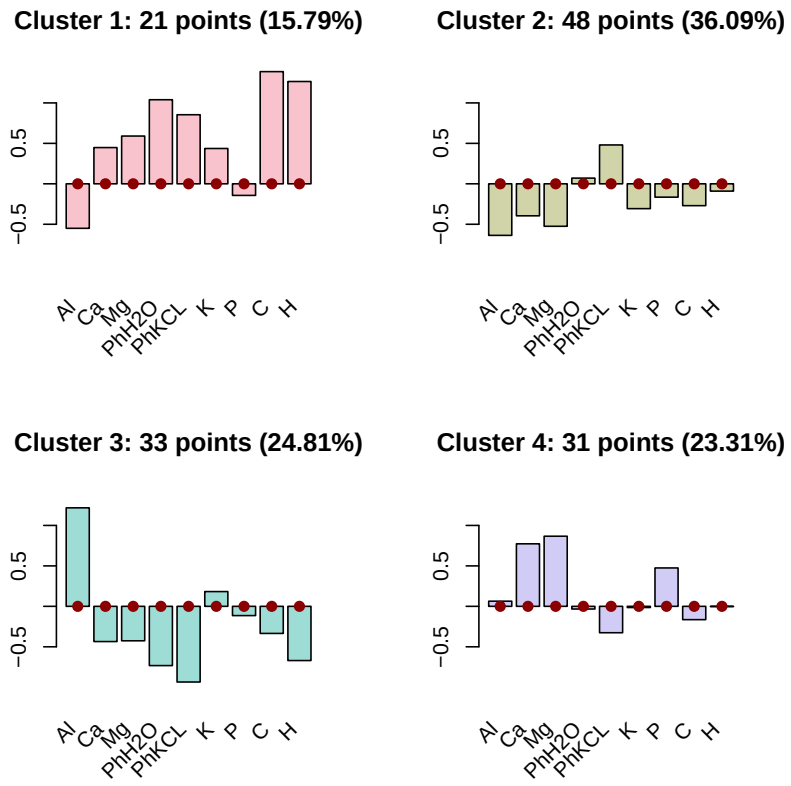
Para avaliar a robustez da média como medida de informação do cluster, repete-se abaixo o procedimento com a mediana.

```
> clust.info.crisp.md <- kcca(DadosClust.p[, -10], centroid, family = kccaFamily("kmedians"))
> clust.info.crisp.md

kcca object of family 'kmedians'

call:
kcca(x = DadosClust.p[, -10], k = centroid, family = kccaFamily("kmedians"))

cluster sizes:

 1  2  3  4
19 48 32 34

> clust.crisp.md <- predict(clust.info.crisp.md)
```

Para entender melhor a característica de cada cluster, na sequência é apresentado, através de um gráfico de barras, os centróides de cada cluster.

```
> table(clust.crisp, clust.crisp.md)

           clust.crisp.md
clust.crisp  1  2  3  4
          1 15  4  1  1
          2  1 44  1  2
          3  0  0 30  3
          4  3  0  0 28
```

```
> barplot(clust.info.crisp)
```



3 Distribuição Espacial dos Agrupamentos

3.1 Distribuição Espacial dos Agrupamentos Nebulosos

O mesmo processo anterior é repetido para os clusters nebulosos.

```
> coord <- dados.ori[, 1:2]
> DadosClustGeo.fuz <- data.frame(coord, clust.fuz)
> head(DadosClustGeo.fuz)
```

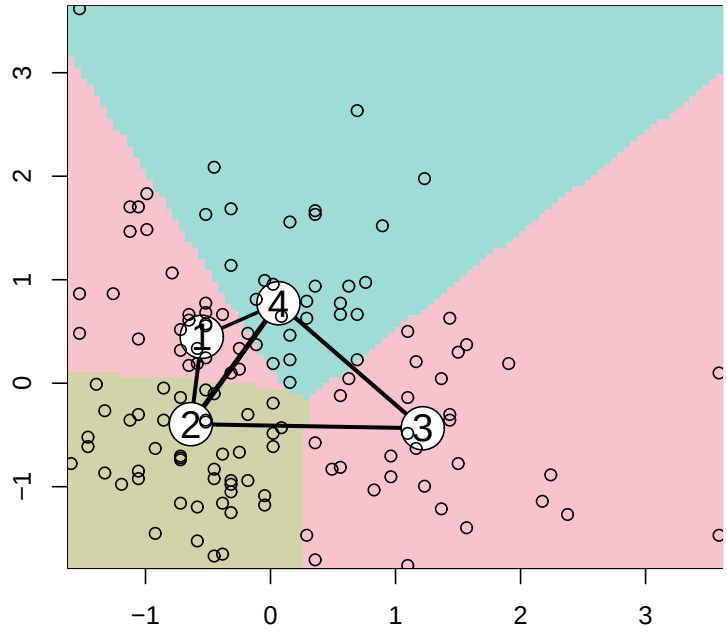
```
       X       Y clust.fuz
1 688713 7190023         1
2 688694 7190000         1
3 688457 7189578         2
4 688790 7190115         3
5 688592 7189756         4
6 688601 7189843         4
```

Carregando o pacote geoR

```
> require(geoR)
```

```
---------------------------------------------------------------
Analysis of geostatistical data
For an Introduction to geoR go to http://www.est.ufpr.br/geoR
geoR version 1.6-13 (built on 2006/12/26) is now loaded
---------------------------------------------------------------

[1] TRUE
```

```
> image(clust.info.crisp)
> points(DadosClust.p[, -10])
```




```
> ViewClust.fuz <- as.geodata(DadosClustGeo.fuz, coords.col = 1:2,
+     dados.col = 3)
```

3.2 Distribuição Espacial dos Agrupamentos Crisp

Abaixo é criado um *data frame* com as coordenadas e os respectivos clusters obtidos pelo método das k-médias.

```
> coord <- dados.ori[, 1:2]
> DadosClustGeo <- data.frame(coord, clust.crisp)
> head(DadosClustGeo)
```

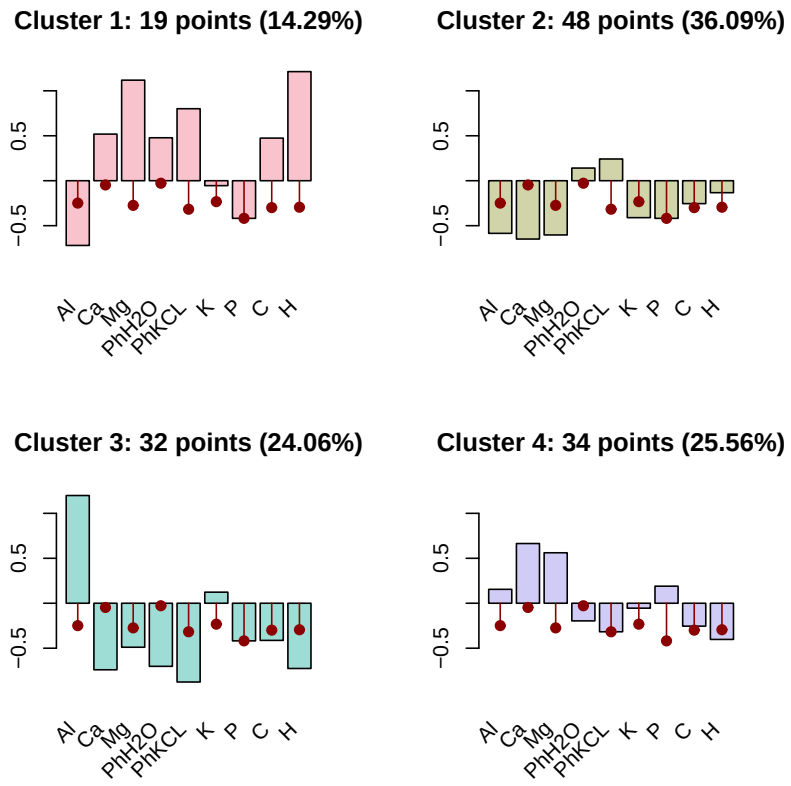
```
       X       Y clust.crisp
1 688713 7190023           4
2 688694 7190000           4
3 688457 7189578           2
4 688790 7190115           3
5 688592 7189756           4
6 688601 7189843           4
```

```
> require(geoR)
```

```
[1] TRUE
```

```
> ViewClust <- as.geodata(DadosClustGeo, coords.col = 1:2, dados.col = 3)
```

```
> barplot(clust.info.crisp.md)
```



4 Extração das Regras de Clusterização

Nesta seção, as classes determinadas pela análise de cluster serão assumidas como verdadeiras e utilizadas como níveis de uma variável dependente em uma árvore de classificação.

Com este procedimento, o objetivo é extrair as regras lógicas que determinaram a formação dos agrupamentos.

Carregando os pacotes para a construção de árvores de classificação.

```
> require(tree)

[1] TRUE

> require(maptree)

[1] TRUE
```

4.1 Extração das Regras para Agrupamentos Nebulosos

```
> d.fuz <- data.frame(DadosClust, Cluster = clust.fuz)
> tree.fuz <- rpart(factor(Cluster) ~ ., data = d.fuz)
> tree.fuz

n= 133

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

```
> plot(ViewClust.fuz$coord)
> for (i in 1:numclust) {
+     i.c = ind.c[i]
+     ViewClust.fuz$data <- clust.fuz == i
+     points(ViewClust.fuz, pt.div = "data.proportional", col = i.c,
+         add = T)
+ }
```

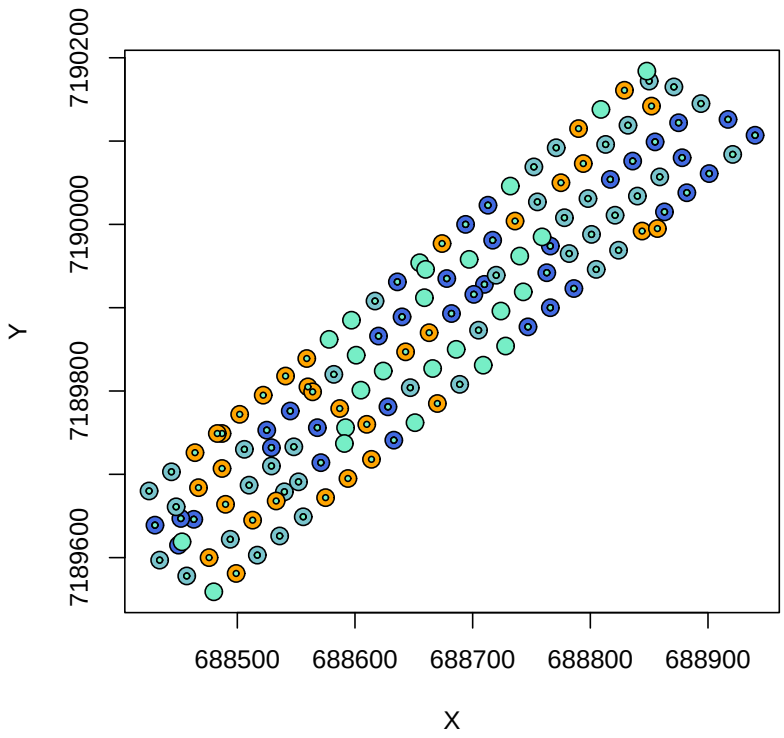
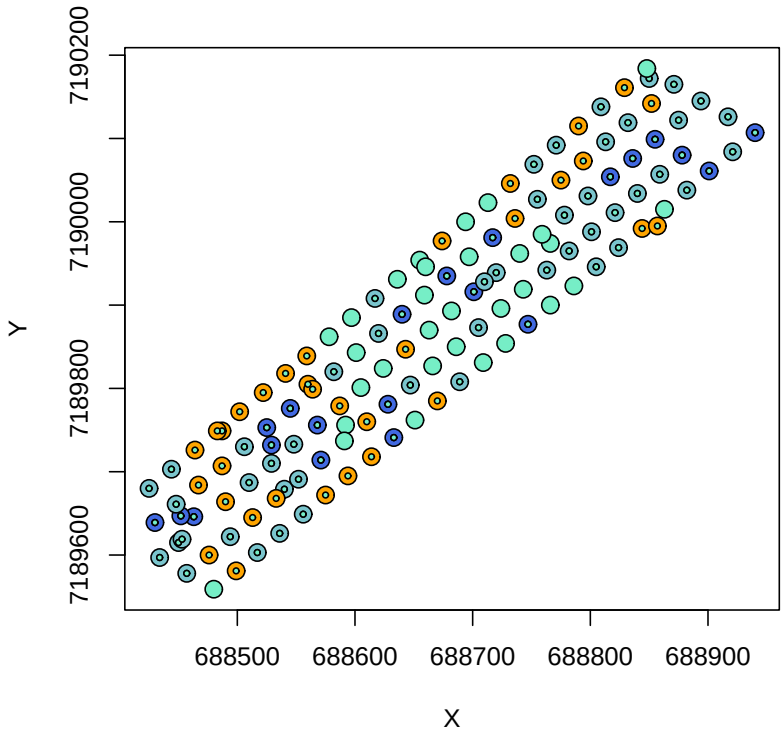


Figura 1: Distribuição Espacial dos Agrupamentos Nebulosos

```
> plot(ViewClust$coord)
> for (i in 1:numclust) {
+     i.c = ind.c[i]
+     ViewClust$data <- clust.crisp == i
+     points(ViewClust, pt.div = "data.proportional", col = i.c,
+         add = T)
+ }
```




```
1) root 133 94 2 (0.27067669 0.29323308 0.24812030 0.18796992)
  2) Al< 3.55 105 66 2 (0.33333333 0.37142857 0.07619048 0.21904762)
    4) Mg< 0.355 58 21 2 (0.18965517 0.63793103 0.10344828 0.06896552)
      8) C>=14.5 7  1 1 (0.85714286 0.14285714 0.00000000 0.00000000) *
      9) C< 14.5 51 15 2 (0.09803922 0.70588235 0.11764706 0.07843137)
       18) PhKCL>=4.225 43  8 2 (0.11627907 0.81395349 0.02325581 0.04651163) *
       19) PhKCL< 4.225 8  3 3 (0.00000000 0.12500000 0.62500000 0.25000000) *
    5) Mg>=0.355 47 23 1 (0.51063830 0.04255319 0.04255319 0.40425532)
      10) PhKCL>=4.375 17  0 1 (1.00000000 0.00000000 0.00000000 0.00000000) *
      11) PhKCL< 4.375 30 11 4 (0.23333333 0.06666667 0.06666667 0.63333333)
        22) H>=5 7  2 1 (0.71428571 0.00000000 0.00000000 0.28571429) *
        23) H< 5 23  6 4 (0.08695652 0.08695652 0.08695652 0.73913043) *
  3) Al>=3.55 28  3 3 (0.03571429 0.00000000 0.89285714 0.07142857) *
```

```
> draw.tree(tree.fuz, nodeinfo = T)
```

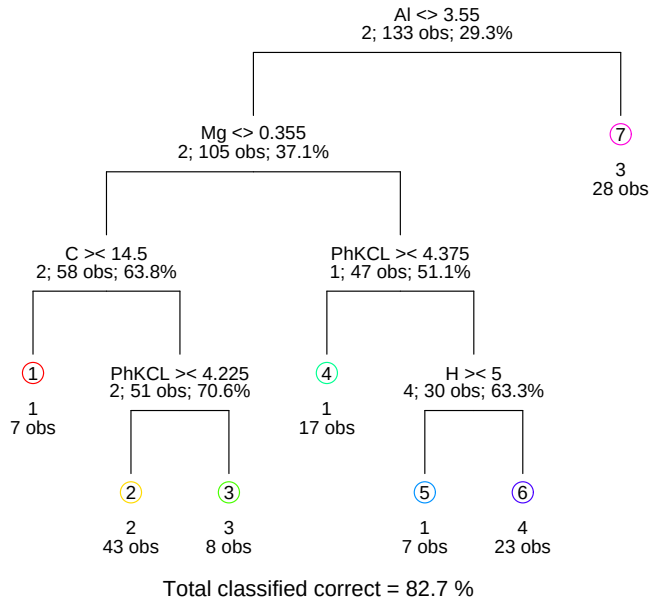


Figura 2: Árvore de Classificação para os Grupos Nebulosos

4.2 Extração das Regras para Agrupamentos Crisp

```
> DadosTreeCrisp <- DadosClust
> DadosTreeCrisp$regiao <- factor(dados.ori$Regiao)
> DadosTreeCrisp$prof <- dados.ori$Prof
> d.crisp <- data.frame(DadosTreeCrisp, Cluster = clust.crisp)
> tree.cl <- rpart(factor(Cluster) ~ ., data = d.crisp)
> tree.cl

n= 133

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

 1) root 133 85 2 (0.15789474 0.36090226 0.24812030 0.23308271)
   2) Al< 3.25 96 48 2 (0.19791667 0.50000000 0.05208333 0.25000000)
     4) Mg< 0.355 54 12 2 (0.12962963 0.77777778 0.07407407 0.01851852)
       8) C>=13.6 7  2 1 (0.71428571 0.28571429 0.00000000 0.00000000) *
       9) C< 13.6 47  7 2 (0.04255319 0.85106383 0.08510638 0.02127660) *
     5) Mg>=0.355 42 19 4 (0.28571429 0.14285714 0.02380952 0.54761905)
      10) C>=12.7 9  0 1 (1.00000000 0.00000000 0.00000000 0.00000000) *
      11) C< 12.7 33 10 4 (0.09090909 0.18181818 0.03030303 0.69696970)
        22) Al< 1.65 11  6 2 (0.27272727 0.45454545 0.00000000 0.27272727) *
        23) Al>=1.65 22  2 4 (0.00000000 0.04545455 0.04545455 0.90909091) *
   3) Al>=3.25 37  9 3 (0.05405405 0.00000000 0.75675676 0.18918919)
     6) Mg< 0.605 30  3 3 (0.06666667 0.00000000 0.90000000 0.03333333) *
     7) Mg>=0.605 7  1 4 (0.00000000 0.00000000 0.14285714 0.85714286) *
```

5 Estatísticas Descritivas dos Clusters

Cluster Crisp x Cluster Fuzz

```
> table(clust.fuz, clust.crisp)
```

```
         clust.crisp
clust.fuz  1  2  3  4
        1 21  7  0  8
        2  0 39  0  0
        3  0  0 32  1
        4  0  2  1 22
```

Cluster x Regiao

```
> table(dados.ori$Regiao, clust.crisp)
```

```
   clust.crisp
     1  2  3  4
  1  7 24 10  8
  2  4  5  2 17
  3 10 19 21  6
```

```
> draw.tree(tree.cl, nodeinfo = T)
```

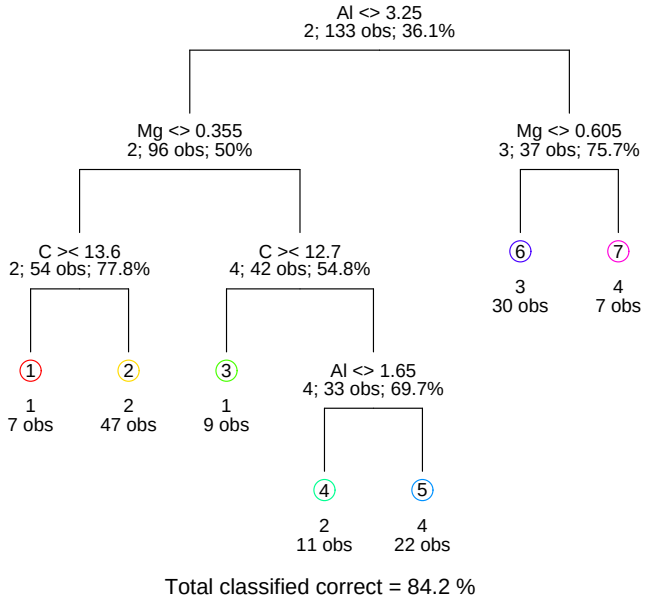


Figura 3: Árvore de Classificação para os Grupos Crisp