Analiza dyskryminacyjna za pomocą metody KNN

Michał Żychowski

2024-01-20

Zaimportowanie naszych danych oraz potrzebnych pakietów

Rozpoczęcie analizy zaczynamy od zaimportowania naszych danych i niezbędnych bibliotek.

```
raw_data <- read.csv('dane.csv',header = TRUE, sep = ',')
library(MASS)
library(ipred)
library(class)
```

Przygotowanie danych

W trakcie analizy nie planujemy wykorzystać wszystkich kolumn dostępnych w bazie danych, zatem teraz dokonamy selekcji tylko tych, które są dla nas istotne.

```
c <- ncol(raw_data)
data <- raw_data[,2:c]
head(data,n = 10)
```

```
##          Size     Weight  Sweetness Crunchiness  Juiciness    Ripeness
## 1  -3.9700485 -2.5123364  5.3463296 -1.01200871  1.8449004  0.32983980
## 2  -1.1952172 -2.8392565  3.6640588  1.58823231  0.8532858  0.86753008
## 3  -0.2920239 -1.3512820 -1.7384292 -0.34261593  2.8386355 -0.03803333
## 4  -0.6571958 -2.2716266  1.3248738 -0.09787472  3.6379705 -3.41376134
## 5   1.3642168 -1.2966119 -0.3846582 -0.55300577  3.0308744 -1.30384943
## 6  -3.4253998 -1.4090822 -1.9135112 -0.55577486 -3.8530715  1.91461592
## 7   1.3316057  1.6359557  0.8759742 -1.67779794  3.1063445 -1.84741673
## 8  -1.9954621 -0.4289585  1.5306436 -0.74297168  0.1588340  0.97443786
## 9  -3.8676322 -3.7345136  0.9864291 -1.20765455  2.2928729  4.08092079
## 10 -0.7279827 -0.4428204 -4.0922228  0.59751292  0.3937143  1.62085677
##       Acidity Quality
## 1  -0.4915905    good
## 2  -0.7228094    good
## 3   2.6216365     bad
## 4   0.7907232    good
## 5   0.5019840    good
## 6  -2.9815232     bad
## 7   2.4141705    good
## 8  -1.4701251    good
## 9  -4.8719048     bad
## 10  2.1856077     bad
```

Jako, że nasze dane są przeskalowane, to nie musimy ich ponownie skalować.

Utworzenie zbioru testowego i uczącego

Obecnie, kiedy posiadamy dane w skali, dokonujemy podziału na zbiór uczący i testowy.

```
indexes<-sample(1:nrow(data),nrow(data)/2,replace = FALSE)
ZU = data[indexes, ]
ZT = data[-indexes, ]
```

Zbiór uczący

```
##            Size     Weight  Sweetness Crunchiness  Juiciness   Ripeness
## 1262  1.2685238  0.1614536 -2.5791522   1.9789730  0.8420687 -0.6090964
## 2658 -0.4951011 -0.4210460  0.2641098   0.2956750  2.0227355  1.0704982
## 2054  0.3025761 -2.0117666  1.9001431   0.5330897  1.7032853  1.2613600
## 3791 -0.7947739 -0.8472749 -0.1246496   1.6250889  1.6066632 -3.2010613
## 3096 -1.0536271  0.1683109  0.1088945   2.5259750 -2.1083369  0.1318201
## 2105 -0.7365113 -1.4656018  0.2104337  -0.1569046  0.5144158 -1.7542743
##          Acidity Quality
## 1262  0.62534563    good
## 2658 -0.04058504    good
## 2054 -0.44960532     bad
## 3791  0.25396609    good
## 3096  1.23964108     bad
## 2105  0.63841881    good
```

Zbiór testowy

```
##          Size     Weight  Sweetness Crunchiness  Juiciness    Ripeness
## 3  -0.2920239 -1.3512820 -1.7384292 -0.34261593  2.8386355 -0.03803333
## 4  -0.6571958 -2.2716266  1.3248738 -0.09787472  3.6379705 -3.41376134
## 5   1.3642168 -1.2966119 -0.3846582 -0.55300577  3.0308744 -1.30384943
## 6  -3.4253998 -1.4090822 -1.9135112 -0.55577486 -3.8530715  1.91461592
## 8  -1.9954621 -0.4289585  1.5306436 -0.74297168  0.1588340  0.97443786
## 10 -0.7279827 -0.4428204 -4.0922228  0.59751292  0.3937143  1.62085677
##       Acidity Quality
## 3   2.6216365     bad
## 4   0.7907232    good
## 5   0.5019840    good
## 6  -2.9815232     bad
## 8  -1.4701251    good
## 10  2.1856077     bad
```

Tworzenie naszego modelu

Obecnie, posiadając już nasze zbiory, możemy stworzyć model dla zbioru uczącego. Tworzymy modele dla 2 i 3 sąsiadów.

```
K1 <- 2
K2 <- 3

KNN_for_2 <- ipredknn(ZU[,8]~.,data=ZU,k=K1)
KNN_for_3 <- ipredknn(ZU[,8]~.,data=ZU,k=K2)
```

Obecnie możemy ocenić jakość dopasowania naszych modeli.

Dla 2 sąsiadów

```
GOF2 <- data.frame(KNN_for_2$learn)
T1 <- table(GOF2$y,GOF2$X.Quality)
T1
```

```
##
##            0    1
##   bad   1001    0
##   good     0  999
```

Procent dobroci dopasowania

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

Dla 3 sąsiadów

```
GOF3 <- data.frame(KNN_for_3$learn)
T2 <- table(GOF3$y,GOF3$X.Quality)
T2
```

```
##
##            0    1
##   bad   1001    0
##   good     0  999
```

Procent dobroci dopasowania

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

Przewidujemy na zbiorze testowym

Dla 2 sąsiadów

```
pred1 <- predict(KNN_for_2,ZT,"class")
S1 <- data.frame(pred1,ZT[,8])
```

Dobroć dopasowania

```
##           prawdziwe
## predykacja bad good
##       bad  925   55
##       good  70  950
```

```
## [1] 0.9375
```

Określimy także prawdopodobieństwo poprawnej oceny obserwacji.

```
pred_p_1<-predict(KNN_for_2,ZT,"prob")
S2<-data.frame(pred_p_1)
```

Wyniki naszej predykcji

```
FS1<-cbind(S1,S2)
colnames(FS1)<-c("Predykcja","Rzeczywistość","Prawdopodobieństwo dla predykcji")
head(FS1, n = 30)
```

```
##    Predykcja Rzeczywistość Prawdopodobieństwo dla predykcji
## 1        bad           bad                              1.0
## 2       good          good                              1.0
## 3       good          good                              1.0
## 4        bad           bad                              1.0
## 5       good          good                              1.0
## 6        bad           bad                              1.0
## 7        bad           bad                              1.0
## 8       good          good                              1.0
## 9       good          good                              1.0
## 10       bad           bad                              1.0
## 11      good          good                              1.0
## 12       bad           bad                              1.0
## 13       bad           bad                              1.0
## 14      good          good                              1.0
## 15       bad           bad                              1.0
## 16       bad           bad                              1.0
## 17       bad           bad                              1.0
## 18       bad           bad                              1.0
## 19      good          good                              1.0
## 20      good           bad                              1.0
## 21      good          good                              1.0
## 22      good          good                              1.0
## 23       bad           bad                              1.0
## 24      good          good                              1.0
## 25      good           bad                              1.0
## 26       bad          good                              1.0
## 27      good          good                              1.0
## 28       bad           bad                              1.0
## 29      good          good                              1.0
## 30      good          good                              0.5
```

Dla 3 sąsiadów

```
pred2 <- predict(KNN_for_3,ZT,"class")
S3 <- data.frame(pred2,ZT[,8])
```

Dobroć dopasowania

```
##           prawdziwe
## predykacja bad good
##       bad  925   55
##       good  70  950
```

```
## [1] 0.9375
```

Określimy również prawdopodobieństwo, że ocena obserwacji została dokładnie określona.

```
pred_p_2<-predict(KNN_for_3,ZT,"prob")
S4<-data.frame(pred_p_2)
```

Wyniki naszej predykcji

```
FS2<-cbind(S3,S4)
colnames(FS2)<-c("Predykcja","Rzeczywistość","Prawdopodobieństwo dla predykcji")
head(FS2, n=30)
```

```
##    Predykcja Rzeczywistość Prawdopodobieństwo dla predykcji
## 1        bad           bad                        1.0000000
## 2       good          good                        1.0000000
## 3       good          good                        1.0000000
## 4        bad           bad                        1.0000000
## 5       good          good                        1.0000000
## 6        bad           bad                        1.0000000
## 7        bad           bad                        1.0000000
## 8       good          good                        1.0000000
## 9       good          good                        1.0000000
## 10       bad           bad                        1.0000000
## 11      good          good                        1.0000000
## 12       bad           bad                        1.0000000
## 13       bad           bad                        1.0000000
## 14      good          good                        1.0000000
## 15       bad           bad                        0.6666667
## 16       bad           bad                        1.0000000
## 17       bad           bad                        0.6666667
## 18       bad           bad                        1.0000000
## 19      good          good                        1.0000000
## 20      good           bad                        0.6666667
## 21      good          good                        0.6666667
## 22      good          good                        1.0000000
## 23       bad           bad                        0.6666667
## 24      good          good                        1.0000000
## 25      good           bad                        0.6666667
```

```
## 26       bad          good                      0.6666667
## 27      good          good                      1.0000000
## 28       bad           bad                      1.0000000
## 29      good          good                      1.0000000
## 30      good          good                      0.6666667
```

Podsumowanie

Na podstawie powyższych działań stwierdzamy, że dla naszych danych liczba dwóch sąsiadów generuje podobne rezultaty jak trzech sąsiadów. W związku z tym liczba 2 sąsiadów jest wystarczająca i nie wymaga zwiększania.