LAB4

Ewa Bojke

15 marca 2020

Zadanie 1

Wyświetlmy ramke z danymi testowymi

```
x=c(2.7,5,7,9,2)
y=c(6,7,3.5,3,5)
Species=c("versicolor","virginica","versicolor","virginica","setosa")
testData=data.frame(x,y,Species)
testData
```

```
##     x   y    Species
## 1 2.7 6.0 versicolor
## 2 5.0 7.0  virginica
## 3 7.0 3.5 versicolor
## 4 9.0 3.0  virginica
## 5 2.0 5.0     setosa
```

```
x=c(1,1,1,2,2,4,5,5,5,6,6,6,6,7,7,8,8)
y=c(4,6,7,2,3,6,4,5,8,3,4,6,7,4,6,2,5)
Species=c("setosa","setosa","setosa","setosa","setosa","versicolor","versicolor","versicolor","virginica","v
ersicolor","versicolor","virginica","virginica","virginica","virginica","versicolor","virginica")
trainData=data.frame(x,y,Species)
```

Przeskalujmy dane i je wyświetlmy

```
testData[,1:2] = scale(testData[,1:2])
trainData[,1:2]= scale(trainData[,1:2])
testData
```

```
##             x           y    Species
## 1 -0.83456001  0.65737574 versicolor
## 2 -0.04788459  1.25499004  virginica
## 3  0.63618099 -0.83666003 versicolor
## 4  1.32024657 -1.13546718  virginica
## 5 -1.07398296  0.05976143     setosa
```

```
trainData
```

```
##             x           y    Species
## 1  -1.5166420 -0.46368166     setosa
## 2  -1.5166420  0.66240238     setosa
## 3  -1.5166420  1.22544440     setosa
## 4  -1.1073894 -1.58976571     setosa
## 5  -1.1073894 -1.02672368     setosa
## 6  -0.2888842  0.66240238 versicolor
## 7   0.1203684 -0.46368166 versicolor
## 8   0.1203684  0.09936036 versicolor
## 9   0.1203684  1.78848642  virginica
## 10  0.5296210 -1.02672368 versicolor
## 11  0.5296210 -0.46368166 versicolor
## 12  0.5296210  0.66240238  virginica
## 13  0.5296210  1.22544440  virginica
## 14  0.9388736 -0.46368166  virginica
## 15  0.9388736  0.66240238  virginica
## 16  1.3481262 -1.58976571 versicolor
## 17  1.3481262  0.09936036  virginica
```

Zbudujmy klasyfikator k-sąsiadów, dla 1 sąsiada

```
library("ipred")
```

```
## Warning: package 'ipred' was built under R version 3.5.3
```

```
klasyfikatorKNN = ipredknn(Species~x+y,testData, k=1)
predykcja= predict(klasyfikatorKNN, testData, "class")
prawdziwe=testData
prawdziwe=prawdziwe[,3]
tablica=table(predykcja, prawdziwe)
```

Wyświetlmy tablicę błędów

```
tablica
```

```
##             prawdziwe
## predykcja    setosa versicolor virginica
##   setosa          1          0         0
##   versicolor      0          2         0
##   virginica       0          0         2
```

Wyświetlmy jaka jest procentowa wartość poprawnej klasyfikacji

```
poprawnosc<- (sum(diag(tablica)) / sum(tablica))*100
poprawnosc
```

```
## [1] 100
```

Zbudujmy klasyfikator k-sąsiadów, dla 3 sąsiadów

```
klasyfikatorKNN2 = ipredknn(Species~x+y,testData, k=3)
predykcja2= predict(klasyfikatorKNN2, testData, "class")
tablica2=table(predykcja2, prawdziwe)
```

Wyświetlmy tablicę błędów

```
tablica2
```

```
##             prawdziwe
## predykcja2   setosa versicolor virginica
##   setosa          0          0         2
##   versicolor      0          2         0
##   virginica       1          0         0
```

Wyświetlmy jaka jest procentowa wartość poprawnej klasyfikacji

```
poprawnosc_a<- (sum(diag(tablica2)) / sum(tablica2))*100
poprawnosc_a
```

```
## [1] 40
```

Wnioski i obserwacje

Lepiej poradził sobie klayfikator 1 najbliższych sąsiadów na małym zbiorze irysów. Poprawność klasyfikacji dla k=1 wynosi 100% natomiast dla k=3 wynosi tylko 40%.

Zadanie 2

Podzielenie na zbiór treningowy/testowy w proporcjach 67/33.

```
ind <- sample(2,nrow(iris), replace=TRUE, prob=c(0.67,0.33))
ind
```

```
##   [1] 1 2 1 2 2 2 2 1 1 1 1 2 1 1 1 1 1 2 1 2 1 1 2 1 2 1 2 1 2 1 1 1 1 1 1
##  [36] 1 1 2 1 2 1 2 2 2 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 2 2 1 1 1 2 1 1 2 1 2
##  [71] 2 1 1 1 2 1 1 1 1 2 1 2 1 2 2 2 1 1 1 1 2 1 1 2 1 2 2 1 1 1 1 1 2 1 1
## [106] 1 2 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 2 1 2
## [141] 1 2 2 1 1 2 1 2 1 1
```

```
train<- iris[ind==1,]
head(train)
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1           5.1         3.5          1.4         0.2  setosa
## 3           4.7         3.2          1.3         0.2  setosa
## 8           5.0         3.4          1.5         0.2  setosa
## 9           4.4         2.9          1.4         0.2  setosa
## 10          4.9         3.1          1.5         0.1  setosa
## 11          5.4         3.7          1.5         0.2  setosa
```

```
tail(train)
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 141          6.7         3.1          5.6         2.4 virginica
## 144          6.8         3.2          5.9         2.3 virginica
## 145          6.7         3.3          5.7         2.5 virginica
## 147          6.3         2.5          5.0         1.9 virginica
## 149          6.2         3.4          5.4         2.3 virginica
## 150          5.9         3.0          5.1         1.8 virginica
```

```
test<- iris[ind==2,]
head(test)
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 2           4.9         3.0          1.4         0.2  setosa
## 4           4.6         3.1          1.5         0.2  setosa
## 5           5.0         3.6          1.4         0.2  setosa
## 6           5.4         3.9          1.7         0.4  setosa
## 7           4.6         3.4          1.4         0.3  setosa
## 12          4.8         3.4          1.6         0.2  setosa
```

```
tail(test)
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 138          6.4         3.1          5.5         1.8 virginica
## 140          6.9         3.1          5.4         2.1 virginica
## 142          6.9         3.1          5.1         2.3 virginica
## 143          5.8         2.7          5.1         1.9 virginica
## 146          6.7         3.0          5.2         2.3 virginica
## 148          6.5         3.0          5.2         2.0 virginica
```

Przeskalujmy dane i je wyświetlmy

```
test[,1:4] = scale(test[,1:4])
train[,1:4]= scale(train[,1:4])
test
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 2    -0.90905853 -0.05083326 -1.254812322 -1.24740965     setosa
## 4    -1.29984284  0.18808305 -1.195180189 -1.24740965     setosa
## 5    -0.77879710  1.38266458 -1.254812322 -1.24740965     setosa
## 6    -0.25775135  2.09941350 -1.075915924 -0.96554305     setosa
## 7    -1.29984284  0.90483197 -1.254812322 -1.10647635     setosa
## 12   -1.03931997  0.90483197 -1.135548057 -1.24740965     setosa
## 18   -0.64853566  1.14374827 -1.254812322 -1.10647635     setosa
## 20   -0.64853566  1.86049719 -1.195180189 -1.10647635     setosa
## 23   -1.29984284  1.38266458 -1.493340852 -1.24740965     setosa
## 25   -1.03931997  0.90483197 -0.956651659 -1.24740965     setosa
## 27   -0.77879710  0.90483197 -1.135548057 -0.96554305     setosa
## 29   -0.51827422  0.90483197 -1.254812322 -1.24740965     setosa
## 38   -0.90905853  1.38266458 -1.254812322 -1.38834296     setosa
## 40   -0.64853566  0.90483197 -1.195180189 -1.24740965     setosa
## 42   -1.43010428 -1.72324740 -1.314444454 -1.10647635     setosa
## 43   -1.56036571  0.42699935 -1.314444454 -1.24740965     setosa
## 44   -0.77879710  1.14374827 -1.135548057 -0.68367644     setosa
## 53    1.69617019  0.18808305  0.832312319  0.58472328 versicolor
## 56    0.13303296 -0.52866587  0.593783788  0.30285667 versicolor
## 60   -0.51827422 -0.76758217  0.235990993  0.44378997 versicolor
## 61   -0.77879710 -2.43999631 -0.002537538 -0.11994324 versicolor
## 65    0.00277152 -0.28974956  0.057094595  0.30285667 versicolor
## 68    0.26329439 -0.76758217  0.355255258 -0.11994324 versicolor
## 70    0.00277152 -1.24541479  0.235990993  0.02099007 versicolor
## 71    0.39355583  0.42699935  0.772680186  1.00752318 versicolor
## 75    1.04486301 -0.28974956  0.474519523  0.30285667 versicolor
## 80    0.13303296 -1.00649848 -0.002537538 -0.11994324 versicolor
## 82   -0.12748992 -1.48433109  0.116726728 -0.11994324 versicolor
## 84    0.52381726 -0.76758217  0.951576584  0.72565658 versicolor
## 85   -0.25775135 -0.05083326  0.593783788  0.58472328 versicolor
## 86    0.52381726  0.90483197  0.593783788  0.72565658 versicolor
## 91   -0.12748992 -1.00649848  0.534151656  0.16192337 versicolor
## 94   -0.77879710 -1.72324740 -0.121801803 -0.11994324 versicolor
## 96    0.13303296 -0.05083326  0.414887391  0.16192337 versicolor
## 97    0.13303296 -0.28974956  0.414887391  0.30285667 versicolor
## 103   1.95669306 -0.05083326  1.428633644  1.43032309  virginica
## 107  -0.90905853 -1.24541479  0.593783788  0.86658988  virginica
## 108   2.21721593 -0.28974956  1.667162175  1.00752318  virginica
## 117   1.17512444 -0.05083326  1.190105114  1.00752318  virginica
## 133   1.04486301 -0.52866587  1.249737247  1.57125639  virginica
## 135   0.65407870 -1.00649848  1.249737247  0.44378997  virginica
## 138   1.04486301  0.18808305  1.190105114  1.00752318  virginica
## 140   1.69617019  0.18808305  1.130472982  1.43032309  virginica
## 142   1.69617019  0.18808305  0.951576584  1.71218969  virginica
## 143   0.26329439 -0.76758217  0.951576584  1.14845648  virginica
## 146   1.43564732 -0.05083326  1.011208716  1.71218969  virginica
## 148   1.17512444 -0.05083326  1.011208716  1.28938979  virginica
```

```
train
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1    -1.02567512  0.95871712  -1.37416108 -1.34304720     setosa
## 3    -1.50533250  0.28390256  -1.42970998 -1.34304720     setosa
## 8    -1.14558946  0.73377894  -1.31861219 -1.34304720     setosa
## 9    -1.86507553 -0.39091199  -1.37416108 -1.34304720     setosa
## 10   -1.26550381  0.05896438  -1.31861219 -1.47077930     setosa
## 11   -0.66593209  1.40859349  -1.31861219 -1.34304720     setosa
## 13   -1.38541815 -0.16597381  -1.37416108 -1.47077930     setosa
## 14   -1.98498987 -0.16597381  -1.54080778 -1.47077930     setosa
## 15   -0.18627471  2.08340805  -1.48525888 -1.34304720     setosa
## 16   -0.30618905  2.98316079  -1.31861219 -1.08758300     setosa
## 17   -0.66593209  1.85846987  -1.42970998 -1.08758300     setosa
```

```
## 19   -0.30618905  1.63353168  -1.20751439 -1.21531510     setosa
## 21   -0.66593209  0.73377894  -1.20751439 -1.34304720     setosa
## 22   -1.02567512  1.40859349  -1.31861219 -1.08758300     setosa
## 24   -1.02567512  0.50884075  -1.20751439 -0.95985091     setosa
## 26   -1.14558946 -0.16597381  -1.26306329 -1.34304720     setosa
## 28   -0.90576077  0.95871712  -1.31861219 -1.34304720     setosa
## 30   -1.50533250  0.28390256  -1.26306329 -1.34304720     setosa
## 31   -1.38541815  0.05896438  -1.26306329 -1.34304720     setosa
## 32   -0.66593209  0.73377894  -1.31861219 -1.08758300     setosa
## 33   -0.90576077  2.30834624  -1.31861219 -1.47077930     setosa
## 34   -0.54601774  2.53328442  -1.37416108 -1.34304720     setosa
## 35   -1.26550381  0.05896438  -1.31861219 -1.34304720     setosa
## 36   -1.14558946  0.28390256  -1.48525888 -1.34304720     setosa
## 37   -0.54601774  0.95871712  -1.42970998 -1.34304720     setosa
## 39   -1.86507553 -0.16597381  -1.42970998 -1.34304720     setosa
## 41   -1.14558946  0.95871712  -1.42970998 -1.21531510     setosa
## 45   -1.02567512  1.63353168  -1.09641659 -1.08758300     setosa
## 46   -1.38541815 -0.16597381  -1.37416108 -1.21531510     setosa
## 47   -1.02567512  1.63353168  -1.26306329 -1.34304720     setosa
## 48   -1.62524684  0.28390256  -1.37416108 -1.34304720     setosa
## 49   -0.78584643  1.40859349  -1.31861219 -1.34304720     setosa
## 50   -1.14558946  0.50884075  -1.37416108 -1.34304720     setosa
## 51    1.25269742  0.28390256   0.45895254  0.18973797 versicolor
## 52    0.53321136  0.28390256   0.34785475  0.31747007 versicolor
## 54   -0.54601774 -1.74054111   0.07011026  0.06200587 versicolor
## 55    0.65312570 -0.61585018   0.40340365  0.31747007 versicolor
## 57    0.41329701  0.50884075   0.45895254  0.44520216 versicolor
## 58   -1.26550381 -1.51560292  -0.31873203 -0.32119042 versicolor
## 59    0.77304004 -0.39091199   0.40340365  0.06200587 versicolor
## 62   -0.06636037 -0.16597381   0.18120805  0.31747007 versicolor
## 63    0.05355398 -1.96547929   0.07011026 -0.32119042 versicolor
## 64    0.17346832 -0.39091199   0.45895254  0.18973797 versicolor
## 66    0.89295439  0.05896438   0.29230585  0.18973797 versicolor
## 67   -0.42610340 -0.16597381   0.34785475  0.31747007 versicolor
## 69    0.29338267 -1.96547929   0.34785475  0.31747007 versicolor
## 72    0.17346832 -0.61585018   0.07011026  0.06200587 versicolor
## 73    0.41329701 -1.29066474   0.57005034  0.31747007 versicolor
## 74    0.17346832 -0.61585018   0.45895254 -0.06572622 versicolor
## 76    0.77304004 -0.16597381   0.29230585  0.18973797 versicolor
## 77    1.01286873 -0.61585018   0.51450144  0.18973797 versicolor
## 78    0.89295439 -0.16597381   0.62559924  0.57293426 versicolor
## 79    0.05355398 -0.39091199   0.34785475  0.31747007 versicolor
## 81   -0.54601774 -1.51560292  -0.04098754 -0.19345832 versicolor
## 83   -0.18627471 -0.84078836   0.01456136 -0.06572622 versicolor
## 87    0.89295439  0.05896438   0.45895254  0.31747007 versicolor
## 88    0.41329701 -1.74054111   0.29230585  0.06200587 versicolor
## 89   -0.42610340 -0.16597381   0.12565916  0.06200587 versicolor
## 90   -0.54601774 -1.29066474   0.07011026  0.06200587 versicolor
## 92    0.17346832 -0.16597381   0.40340365  0.18973797 versicolor
## 93   -0.18627471 -1.06572655   0.07011026 -0.06572622 versicolor
## 95   -0.42610340 -0.84078836   0.18120805  0.06200587 versicolor
## 98    0.29338267 -0.39091199   0.23675695  0.06200587 versicolor
## 99   -1.02567512 -1.29066474  -0.48537872 -0.19345832 versicolor
## 100  -0.30618905 -0.61585018   0.12565916  0.06200587 versicolor
## 101   0.41329701  0.50884075   1.18108822  1.59479104  virginica
## 102  -0.18627471 -0.84078836   0.68114814  0.82839846  virginica
## 104   0.41329701 -0.39091199   0.95889262  0.70066636  virginica
## 105   0.65312570 -0.16597381   1.06999042  1.21159475  virginica
## 106   1.97218349 -0.16597381   1.51438160  1.08386265  virginica
## 109   0.89295439 -1.29066474   1.06999042  0.70066636  virginica
## 110   1.49252611  1.18365531   1.23663711  1.59479104  virginica
## 111   0.65312570  0.28390256   0.68114814  0.95613055  virginica
## 112   0.53321136 -0.84078836   0.79224593  0.82839846  virginica
## 113   1.01286873 -0.16597381   0.90334373  1.08386265  virginica
## 114  -0.30618905 -1.29066474   0.62559924  0.95613055  virginica
## 115  -0.18627471 -0.61585018   0.68114814  1.46705894  virginica
## 116   0.53321136  0.28390256   0.79224593  1.33932685  virginica
## 118   2.09209783  1.63353168   1.56993050  1.21159475  virginica
## 119   2.09209783 -1.06572655   1.68102830  1.33932685  virginica
## 120   0.05355398 -1.96547929   0.62559924  0.31747007  virginica
## 121   1.13278308  0.28390256   1.01444152  1.33932685  virginica
## 122  -0.42610340 -0.61585018   0.57005034  0.95613055  virginica
## 123   2.09209783 -0.61585018   1.56993050  0.95613055  virginica
```

```
## 124    0.41329701 -0.84078836   0.57005034  0.70066636  virginica
## 125    0.89295439  0.50884075   1.01444152  1.08386265  virginica
## 126    1.49252611  0.28390256   1.18108822  0.70066636  virginica
## 127    0.29338267 -0.61585018   0.51450144  0.70066636  virginica
## 128    0.17346832 -0.16597381   0.57005034  0.70066636  virginica
## 129    0.53321136 -0.61585018   0.95889262  1.08386265  virginica
## 130    1.49252611 -0.16597381   1.06999042  0.44520216  virginica
## 131    1.73235480 -0.61585018   1.23663711  0.82839846  virginica
## 132    2.33192652  1.63353168   1.40328381  0.95613055  virginica
## 134    0.41329701 -0.61585018   0.68114814  0.31747007  virginica
## 136    2.09209783 -0.16597381   1.23663711  1.33932685  virginica
## 137    0.41329701  0.73377894   0.95889262  1.46705894  virginica
## 139    0.05355398 -0.16597381   0.51450144  0.70066636  virginica
## 141    0.89295439  0.05896438   0.95889262  1.46705894  virginica
## 144    1.01286873  0.28390256   1.12553932  1.33932685  virginica
## 145    0.89295439  0.50884075   1.01444152  1.59479104  virginica
## 147    0.41329701 -1.29066474   0.62559924  0.82839846  virginica
## 149    0.29338267  0.73377894   0.84779483  1.33932685  virginica
## 150   -0.06636037 -0.16597381   0.68114814  0.70066636  virginica
```

```
myFormula<-Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width
```

Zbudujmy klasyfikator k-sąsiadów, dla 1 sąsiada

```
library("ipred")
klasyfikatorKNN = ipredknn(myFormula,test, k=1)
predykcja= predict(klasyfikatorKNN, test, "class")
prawdziwe=test
prawdziwe=prawdziwe[,5]
tablica=table(predykcja, prawdziwe)
```

Wyświetlmy tablicę błędów

```
tablica
```

```
##             prawdziwe
## predykcja    setosa versicolor virginica
##   setosa         17          0         0
##   versicolor      0         18         0
##   virginica       0          0        12
```

Wyświetlmy jaka jest procentowa wartość poprawnej klasyfikacji

```
poprawnosc<- (sum(diag(tablica)) / sum(tablica))*100
poprawnosc
```

```
## [1] 100
```

Zbudujmy klasyfikator k-sąsiadów, dla 3 sąsiadów

```
library("ipred")
klasyfikatorKNN3 = ipredknn(myFormula,test, k=3)
predykcja= predict(klasyfikatorKNN3, test, "class")
prawdziwe=test
prawdziwe=prawdziwe[,5]
tablica3=table(predykcja, prawdziwe)
```

Wyświetlmy tablicę błędów

```
tablica3
```

```
##             prawdziwe
## predykcja    setosa versicolor virginica
##   setosa         17          0         0
##   versicolor      0         16         1
##   virginica       0          2        11
```

Wyświetlmy jaka jest procentowa wartość poprawnej klasyfikacji

```
poprawnosc3<- (sum(diag(tablica3)) / sum(tablica3))*100
poprawnosc3
```

```
## [1] 93.61702
```

Zbudujmy klasyfikator k-sąsiadów, dla 5 sąsiadów

```
library("ipred")
klasyfikatorKNN5 = ipredknn(myFormula,test, k=5)
predykcja= predict(klasyfikatorKNN5, test, "class")
prawdziwe=test
prawdziwe=prawdziwe[,5]
tablica5=table(predykcja, prawdziwe)
```

Wyświetlmy tablicę błędów

```
tablica5
```

```
##             prawdziwe
## predykcja    setosa versicolor virginica
##   setosa         17          0         0
##   versicolor      0         17         2
##   virginica       0          1        10
```

Wyświetlmy jaka jest procentowa wartość poprawnej klasyfikacji

```
poprawnosc5<- (sum(diag(tablica5)) / sum(tablica5))*100
poprawnosc5
```

```
## [1] 93.61702
```

Zbudujmy klasyfikator k-sąsiadów, dla 11 sąsiadów

```
library("ipred")
klasyfikatorKNN11 = ipredknn(myFormula,test, k=11)
predykcja= predict(klasyfikatorKNN11, test, "class")
prawdziwe=test
prawdziwe=prawdziwe[,5]
tablica11=table(predykcja, prawdziwe)
```

Wyświetlmy tablicę błędów

```
tablica11
```

```
##             prawdziwe
## predykcja    setosa versicolor virginica
##   setosa         16          0         0
##   versicolor      1         17         3
##   virginica       0          1         9
```

Wyświetlmy jaka jest procentowa wartość poprawnej klasyfikacji

```
poprawnosc11<- (sum(diag(tablica11)) / sum(tablica11))*100
poprawnosc11
```

```
## [1] 89.3617
```

Wnioski i interpretacje

Im więcej sąsiadów bierzemy pod uwagę tym nasze wyniki poprawności klasyfikacji są gorsze. Najlepiej patrzeć na jak najmnieszą ilość sąsiadów aby nasze wyniki były jak najbardziej wiarygodne. Dla k=1 uzyskujemy 100% poprawność. Jest to najlpeszy klasyfikator.