ml

March 7, 2023

Rescaling with z-core standardisation

- Get the data and transform the dataframe for classification:

```
## get the Wisconsin breast cancer data as data frame:
wbcd <- read.csv(file="http://bit.ly/3khqmkp")
## drop the first (ID) column:
wbcd <- wbcd[-1]
## recode target class as labeled 2-level factor
wbcd$diagnosis |> factor(c("B","M"),c("Benign","Malignant")) -> wbcd$diagnosis
# wbcd$diagnosis |>  str()

str(wbcd$diagnosis)

Factor w/ 2 levels "Benign","Malignant": 1 1 1 1 1 1 1 2 1 1 ...
```

- Problem: Use the z-score standardization to transform the data, check and interpret the predictions.
- This normalization method is well suited to a cancer dataset:
  - z-score standardized values have no predefined minimum or maximum
  - extreme values are not compressed towards the center
  - a malignant, fast-growing tumor might lead to extreme outliers
  - outliers weigh more heavily in the distance calculation
- To standardize a vector, use `base::scale`, which centers and scales the columns of a `numeric matrix`:

	wbcd_test_pred		
wbcd_test_labels	Benign	Malignant	Row Total
Benign	61	0	61
	1.000	0.000	0.610
	0.924	0.000	
	0.610	0.000	
Malignant	5	34	39
	0.128	0.872	0.390
	0.076	1.000	
	0.050	0.340	
Column Total	66	34	100
	0.660	0.340	

Figure 1: Sample results after z-score standardization

```
args(scale)

function (x, center = TRUE, scale = TRUE)
NULL
```

- `scale` can be directly applied to a data frame so there is no need to use `lapply`:

```
wbcd_z <- as.data.frame(scale(x=wbcd[-1])) # exclude factor
```

- Check the transformation with `summary` on `area_mean`, which showed a huge spread in values:

```
summary(wbcd$area_mean)    # original feature
summary(wbcd_z$area_mean)  # after z-score standardization
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  143.5   420.3   551.1   654.9   782.7  2501.0
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-1.4532 -0.6666 -0.2949  0.0000  0.3632  5.2459
```

- The `mean` of a z-score standardized variable should always be (close to) 0, and the range should be compact (use `diff`):

```
mean(wbcd$area_mean)
diff(range(wbcd$area_mean))
mean(wbcd_z$area_mean)
diff(range(wbcd_z$area_mean))
```

```
[1] 654.8891
[1] 2357.5
[1] 1.219424e-16
[1] 6.699077
```

- To see the effect of this transformation:

  1. split the standardized data in training and test data
  2. define training and test labels (classes)
  3. run `knn` with `k=21` on the data

4. evaluate the performance with the confidence matrix

```
library(class)
library(gmodels)
wbcd_train <- wbcd_z[1:469, ]
wbcd_test <- wbcd_z[470:569, ]
wbcd_train_labels <- wbcd[1:469, 1]
wbcd_test_labels <- wbcd[470:569, 1]
wbcd_test_pred <- knn(train = wbcd_train,
                      test = wbcd_test,
                      cl = wbcd_train_labels,
                      k = 21)
CrossTable(x = wbcd_test_labels,
           y = wbcd_test_pred,
           prop.chisq = FALSE)
```

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  100


 | wbcd_test_pred
wbcd_test_labels |    Benign | Malignant | Row Total |
-----------------|-----------|-----------|-----------|
  Benign |        61 |         0 |        61 |
 |     1.000 |     0.000 |     0.610 |
 |     0.924 |     0.000 |           |
 |     0.610 |     0.000 |           |
-----------------|-----------|-----------|-----------|
       Malignant |         5 |        34 |        39 |
 |     0.128 |     0.872 |     0.390 |
```

```
 |     0.076 |     1.000 |           |
 |     0.050 |     0.340 |           |
-----------------|-----------|-----------|-----------|
    Column Total |        66 |        34 |       100 |
 |     0.660 |     0.340 |           |
-----------------|-----------|-----------|-----------|

Total Observations in Table:  100

                 | wbcd_test_pred
wbcd_test_labels |    Benign | Malignant | Row Total |
-----------------|-----------|-----------|-----------|
          Benign |        61 |         0 |        61 |
                 |     1.000 |     0.000 |     0.610 |
                 |     0.924 |     0.000 |           |
                 |     0.610 |     0.000 |           |
-----------------|-----------|-----------|-----------|
       Malignant |         5 |        34 |        39 |
                 |     0.128 |     0.872 |     0.390 |
                 |     0.076 |     1.000 |           |
                 |     0.050 |     0.340 |           |
-----------------|-----------|-----------|-----------|
    Column Total |        66 |        34 |       100 |
                 |     0.660 |     0.340 |           |
-----------------|-----------|-----------|-----------|
```

- Unfortunately, these values are worse than after the min-max normalization: the number of false negatives has increased from 2 to 5.