Research Log Project Machine Learning

Diagnosing malignancy of breast masses using Machine Learning

Student: Vincent Talen

Student number: 389015

Class: BFV3

Study: Bioinformatics

Institute: Institute for Life Science & Technology

Teachers: Dave Langers (LADR) and Bart Barnard (BABA)

Date: 2022-09-22

Contents

1 Preparing R environment **2**

2 Data Set **3**

2.1 Origin of the data 3

2.2 Collection of the data 3

2.3 Data structure 3

2.4 Loading in the data 4

1 Preparing R environment

For the data analysis and further processes multiple libraries are needed, they are loaded in here. A few other options/settings are also configured here.

```r
# Create vector with all packages that are required
packages <- c("readr", "pander", "ggplot2")
# Load each package in the vector with lapply
invisible(lapply(packages, library, character.only = TRUE))
# Drop the packages variable from memory since it will not be used again
remove(packages)

# Disable printing 'table continues' lines between split sections of pander tables
panderOptions("table.continues", "")
```

2 Data Set

2.1 Origin of the data

The data set that is used is the *Wisconsin Breast Cancer (Diagnostic) Data Set*, which is publicly available from the UCI Machine Learning Repository. There are two published research articles, from the same team of researchers, where the data set was first used, namely [1] and [2]. The samples for the data were collected from 569 patients at the University of Wisconsin Hospital.

2.2 Collection of the data

The data was gathered by first collecting the fine needle aspirates (FNA), which are expressed on a glass slide and stained. A color video camera mounted on top of a microscope, where the images were projected into the camera with a 63x objective and 2.5x ocular. The image was then captured by a color frame grabber board as a 512x480, 8-bit-per-pixel Targa file.
The digitized image is then analyzed in the program Xcyt (custom made by Nick Street). First the user marks approximate initial boundaries of the nuclei and then the actual boundaries are further defined with an active contour model known as "Snake". In the end the snake reaches a point where it's curve accurately corresponds to the boundary of a cell nucleus. From the snake-generated cell nuclei boundaries 10 features are extracted, these are numerically modeled such that larger values will typically indicate a higher likelihood of malignancy.

The ten features that are extracted for each cell nucleus are the following:

1. Radius (mean of distances from center to points on the perimeter)
2. Texture (standard deviation of gray-scale values)
3. Perimeter (the total distance between all the points of the snake-generated boundary)
4. Area (the nuclear area is the sum of pixels on the interior, with half of the pixels of the perimeter)
5. Smoothness (local variation in radius lengths)
6. Compactness (perimeter^2 / area - 1.0)
7. Concavity (severity of concave portions of the contour)
8. Concave points (number of concave portions of the contour)
9. Symmetry (difference in length of perpendicular lines to the longest chord through the center, in both directions)
10. Fractal dimension ("coastline approximation" - 1)

For each feature for every image, three final values were computed and saved to the data set, namely the mean, standard error and the extreme (largest) value.

2.3 Data structure

FNA samples were taken from 569 patients, resulting in a data set with features of nuclei boundaries from 569 images and 32 columns. An ID column, a column with the diagnosis (benign or malignant) and 30 columns of the features describing the nuclei boundaries (10x mean/extreme/se).
Because the data set itself does not come with an annotated header with column names, a codebook has been manually made. This codebook has the abbreviated column name, the full column name, the data type and a description for each feature/column.

Below is an overview of the columns in the data set, shown using the contents of the codebook after it has been loaded in:

```r
# Import codebook
codebook <- read_delim("data/codebook.txt", delim = "|", show_col_types = FALSE)
# Pretty print the summary of the codebook
pander::pander(codebook[,1:3], style = "rmarkdown")
```

Column Name	Full Name	Type
id	ID	dbl
diagnosis	Diagnosis	fct
radius_mean	Mean Radius	dbl
texture_mean	Mean Texture	dbl
perimeter_mean	Mean Perimeter	dbl
area_mean	Mean Area	dbl
smoothness_mean	Mean Smoothness	dbl
compactness_mean	Mean Compactness	dbl
concavity_mean	Mean Concavity	dbl
concave_pts_mean	Mean Concave Points	dbl
symmetry_mean	Mean Symmetry	dbl
fractal_dim_mean	Mean Fractal Dimension	dbl
radius_se	Radius Standard Error	dbl
texture_se	Texture Standard Error	dbl
perimeter_se	Perimeter Standard Error	dbl
area_se	Area Standard Error	dbl
smoothness_se	Smoothness Standard Error	dbl
compactness_se	Compactness Standard Error	dbl
concavity_se	Concavity Standard Error	dbl
concave_pts_se	Concave Points Standard Error	dbl
symmetry_se	Symmetry Standard Error	dbl
fractal_dim_se	Fractal Dimension Standard Error	dbl
radius_worst	Worst Radius	dbl
texture_worst	Worst Texture	dbl
perimeter_worst	Worst Perimeter	dbl
area_worst	Worst Area	dbl
smoothness_worst	Worst Smoothness	dbl
compactness_worst	Worst Compactness	dbl
concavity_worst	Worst Concavity	dbl
concave_pts_worst	Worst Concave Points	dbl
symmetry_worst	Worst Symmetry	dbl
fractal_dim_worst	Worst Fractal Dimension	dbl

As can be seen, all the features are of the type double except the main classification factor column.

2.4 Loading in the data

```r
# Load in data from file with codebook column names
data <- read_csv("data/wdbc.data", col_names = codebook[[1]], show_col_types = FALSE)
# Set diagnosis column to factor
data$diagnosis <- factor(data$diagnosis, labels = c("Benign", "Malignant"))

# Print data to check if it is loaded in correctly
pander::pander(head(data[,c(1:6, 12:16, 22:26)]))
```

id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean
842302	Malignant	17.99	10.38	122.8	1001
842517	Malignant	20.57	17.77	132.9	1326
84300903	Malignant	19.69	21.25	130	1203
84348301	Malignant	11.42	20.38	77.58	386.1
84358402	Malignant	20.29	14.34	135.1	1297
843786	Malignant	12.45	15.7	82.57	477.1

fractal_dim_mean	radius_se	texture_se	perimeter_se	area_se
0.07871	1.095	0.9053	8.589	153.4
0.05667	0.5435	0.7339	3.398	74.08
0.05999	0.7456	0.7869	4.585	94.03
0.09744	0.4956	1.156	3.445	27.23
0.05883	0.7572	0.7813	5.438	94.44
0.07613	0.3345	0.8902	2.217	27.19

fractal_dim_se	radius_worst	texture_worst	perimeter_worst	area_worst
0.006193	25.38	17.33	184.6	2019
0.003532	24.99	23.41	158.8	1956
0.004571	23.57	25.53	152.5	1709
0.009208	14.91	26.5	98.87	567.7
0.005115	22.54	16.67	152.2	1575
0.005082	15.47	23.75	103.4	741.6

```r
# Print the amount of samples and columns
cat("Amount of samples:", dim(data)[1], "\tColumns in dataframe", dim(data)[2], "\n")
```

```
## Amount of samples: 569   Columns in dataframe 32
```

```r
# Print diagnosis counts
table(data$diagnosis)
```

```
##
##    Benign Malignant
##       357       212
```

References

[1] W.N. Street, W.H. Wolberg and O.L. Mangasarian. (1993), *Nuclear feature extraction for breast tumor diagnosis.*, 1993 International Symposium on Electronic Imaging: Science and Technology, volume 1905, pages 861-870, https://doi.org/10.1117/12.148698 (accessed Sep 16, 2022).

[2] O.L. Mangasarian, W.N. Street and W.H. Wolberg. (1995), *Breast cancer diagnosis and prognosis via linear programming*, Operations Research, volume 43, issue 4, pages 570-577, https://doi.org/10.1287/opre.43.4.570 (accessed Sep 17, 2022).