Breast Cancer

Durga Gaddam

August 12, 2016

Objective:

The current objective of the article is to Use Machine Learning concept of Supervised Learing and one of the Algorithm form of KNN- k Nearest Neighbour to automate the identification of cancerous cells.

The Data used has been extracted from Wisconsin Breast Cancer Diagonistic dataset

Concept of K-Nearest Neighbour Algorithm:

Algorithm is a sequence of procedures or rules given to a computer, when followed guarantees the result

In KNN algorithm, the data is divided into groups according to the scores given. To estimate the required target group, distance from each group is calculated and the nearest neighbour group is selected with additional conditions.

Step-1: Collecting the Data

Step-2: Exploring and Preparing the data

Step-3: Training a model on the data

Step-4: Evaluating model performance

Step-5: Improving model perfomance

STEP-1 and STEP-2 Loading and Understanding/preparing the data

The dataset consists of 32 columns and 568 observations (patient details)

```r
###install.packages("class")
###install.packages("gmodels")
###library(class)
###library(gmodels)


bcdata <- read.csv("F:/R PRACTICE/Breast Cancer/Breastcancerdata.csv", stringsAsFactors = FALSE)

names(bcdata)
```

```
##  [1] "Id"                         "Diagnosis"
##  [3] "Radius_Mean"                "Texture_Mean"
##  [5] "Perimeter_Mean"             "Area_Mean"
##  [7] "Smoothness_Mean"            "Compactness_Mean"
##  [9] "Concavity_Mean"          "Concave.Points_Mean"
## [11] "Symmetry_Mean"           "Fractal_Dimension_Mean"
## [13] "Radius_Se"               "Texture_Se"
## [15] "Perimeter_Se"            "Area_Se"
## [17] "Smoothness_Se"           "Compactness_Se"
## [19] "Concavity_Se"            "Concave.Points_Se"
## [21] "Symmetry_Se"             "Fractal_Dimension_Se"
## [23] "Radius_Worst"            "Texture_Worst"
## [25] "Perimeter_Worst"         "Area_Worst"
## [27] "Smoothness_Worst"        "Compactness_Worst"
## [29] "Concavity_Worst"         "Concave.Points_Worst"
## [31] "Symmetry_Worst"          "Fractal_Dimension_Worst"
```

The First column is ID of the cell and is of no use for understanding and using in Machine learning. We need to remove ID column to avoid overfitting.

The second column indicates the result of Diagonsis as “M” for Malignent and “B” for Benign. This data is a categorical data and should be converted into nominal data.

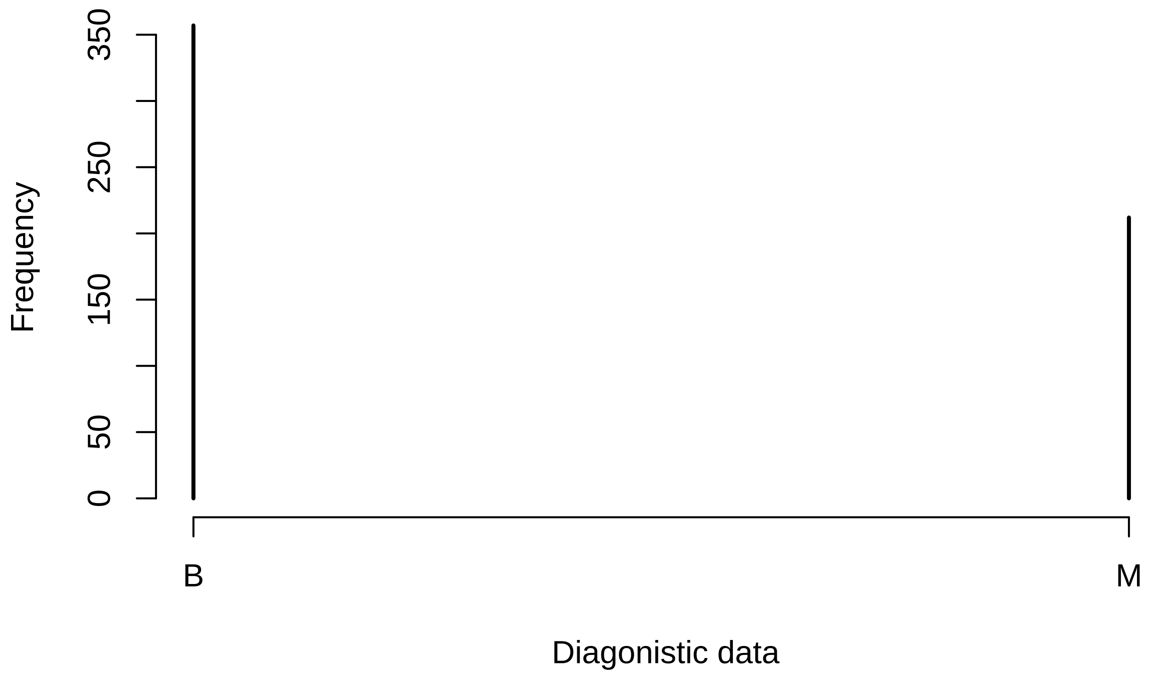
```r
str(bcdata)
```

```
## 'data.frame':    569 obs. of  32 variables:
##  $ Id                     : int  842302 842517 84300903 84348301 84358402 843786 844359 84458202 8449
##  $ Diagnosis              : chr  "M" "M" "M" "M" ...
##  $ Radius_Mean            : num  18 20.6 19.7 11.4 20.3 ...
##  $ Texture_Mean           : num  10.4 17.8 21.2 20.4 14.3 ...
##  $ Perimeter_Mean         : num  122.8 132.9 130 77.6 135.1 ...
##  $ Area_Mean              : num  1001 1326 1203 386 1297 ...
##  $ Smoothness_Mean        : num  0.1184 0.0847 0.1096 0.1425 0.1003 ...
##  $ Compactness_Mean       : num  0.2776 0.0786 0.1599 0.2839 0.1328 ...
##  $ Concavity_Mean         : num  0.3001 0.0869 0.1974 0.2414 0.198 ...
##  $ Concave.Points_Mean    : num  0.1471 0.0702 0.1279 0.1052 0.1043 ...
##  $ Symmetry_Mean          : num  0.242 0.181 0.207 0.26 0.181 ...
##  $ Fractal_Dimension_Mean : num  0.0787 0.0567 0.06 0.0974 0.0588 ...
##  $ Radius_Se              : num  1.095 0.543 0.746 0.496 0.757 ...
##  $ Texture_Se             : num  0.905 0.734 0.787 1.156 0.781 ...
##  $ Perimeter_Se           : num  8.59 3.4 4.58 3.44 5.44 ...
##  $ Area_Se                : num  153.4 74.1 94 27.2 94.4 ...
##  $ Smoothness_Se          : num  0.0064 0.00522 0.00615 0.00911 0.01149 ...
##  $ Compactness_Se         : num  0.049 0.0131 0.0401 0.0746 0.0246 ...
##  $ Concavity_Se           : num  0.0537 0.0186 0.0383 0.0566 0.0569 ...
##  $ Concave.Points_Se      : num  0.0159 0.0134 0.0206 0.0187 0.0188 ...
##  $ Symmetry_Se            : num  0.03 0.0139 0.0225 0.0596 0.0176 ...
##  $ Fractal_Dimension_Se   : num  0.00619 0.00353 0.00457 0.00921 0.00511 ...
##  $ Radius_Worst           : num  25.4 25 23.6 14.9 22.5 ...
##  $ Texture_Worst          : num  17.3 23.4 25.5 26.5 16.7 ...
##  $ Perimeter_Worst        : num  184.6 158.8 152.5 98.9 152.2 ...
##  $ Area_Worst             : num  2019 1956 1709 568 1575 ...
##  $ Smoothness_Worst       : num  0.162 0.124 0.144 0.21 0.137 ...
##  $ Compactness_Worst      : num  0.666 0.187 0.424 0.866 0.205 ...
##  $ Concavity_Worst        : num  0.712 0.242 0.45 0.687 0.4 ...
##  $ Concave.Points_Worst   : num  0.265 0.186 0.243 0.258 0.163 ...
##  $ Symmetry_Worst         : num  0.46 0.275 0.361 0.664 0.236 ...
##  $ Fractal_Dimension_Worst: num  0.1189 0.089 0.0876 0.173 0.0768 ...
```

```
bcdata <- bcdata[-1]
table(bcdata$Diagnosis)
```

```
##
##   B   M
## 357 212
```

```
plot(table(bcdata$Diagnosis), xlab = " Diagonistic data", ylab = " Frequency")
```



```
## converting the Diagnosis data into factors

bcdata$Diagnosis <- factor(bcdata$Diagnosis, levels=c("B","M"), labels = c("Benign", "Malignant"))
```

Calculating the proprotions of each result

```
round(prop.table(table(bcdata$Diagnosis))*100, digits=1)
```

```
##
##    Benign Malignant
##      62.7      37.3
```

```
summary(bcdata[c("Radius_Mean", "Area_Mean", "Smoothness_Mean")])
```

```
##   Radius_Mean       Area_Mean      Smoothness_Mean
##  Min.   : 6.981   Min.   : 143.5   Min.   :0.05263
##  1st Qu.:11.700   1st Qu.: 420.3   1st Qu.:0.08637
##  Median :13.370   Median : 551.1   Median :0.09587
##  Mean   :14.127   Mean   : 654.9   Mean   :0.09636
##  3rd Qu.:15.780   3rd Qu.: 782.7   3rd Qu.:0.10530
##  Max.   :28.110   Max.   :2501.0   Max.   :0.16340
```

The range of area is 2501-143.5= 2357.5 which is abnormal, this can be rectified by normalizing the data.

Normalization

```
nd <- function(x){

  return((x-min(x))/(max(x)-min(x)))
}

### Applying the function to 31 columns of the dataset excluding Diagnosis column
bcdata_n <- as.data.frame(lapply(bcdata[2:31],nd))

summary(bcdata_n$Area_Mean)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.1174  0.1729  0.2169  0.2711  1.0000
```

```
### Now we can see that data is normalized and the range interval of area is 0 to 1

### we now divide the data into 469 and 100 records to predict the last 100 records

bcdata_train <- bcdata_n[1:469,]
bcdata_test <- bcdata_n[470:569,]

### As we have excluded the Diagonsis column from normalized data, we need to store that column in new

bcdata_train_labels <- bcdata[1:469,1]
bcdata_test_labels <- bcdata[470:569,1]
```

STEP-3 TRAINING THE DATA MODEL

```
require(class)
```

```
## Loading required package: class
```

```
## Warning: package 'class' was built under R version 3.3.1
```

```
bcdata_pred <- knn(train=bcdata_train,test=bcdata_test,cl=bcdata_train_labels,k=21)
```

STEP-4 EVALUATING MODEL PERFORMANCE

```
require(gmodels)
```

```
## Loading required package: gmodels
```

```
## Warning: package 'gmodels' was built under R version 3.3.1
```

```
CrossTable(x=bcdata_test_labels,y=bcdata_pred,prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                    | bcdata_pred
## bcdata_test_labels |    Benign | Malignant | Row Total |
## -------------------|-----------|-----------|-----------|
##             Benign |        77 |         0 |        77 |
##                    |     1.000 |     0.000 |     0.770 |
##                    |     0.975 |     0.000 |           |
##                    |     0.770 |     0.000 |           |
## -------------------|-----------|-----------|-----------|
##          Malignant |         2 |        21 |        23 |
##                    |     0.087 |     0.913 |     0.230 |
##                    |     0.025 |     1.000 |           |
##                    |     0.020 |     0.210 |           |
## -------------------|-----------|-----------|-----------|
##       Column Total |        79 |        21 |       100 |
##                    |     0.790 |     0.210 |           |
## -------------------|-----------|-----------|-----------|
##
##
```

STEP-5 IMPROVING THE MODEL

The model can be improved by using different K values and using z-scores instead of normalization.