**STAT6620-01- Homework2**

**CHINKI RAI**

**04/03/2017**

**Example of Diagnosing breast cancer:**

machine learning could automate the identification of cancerous cells, it would provide considerable benefit to the health system.

**Step 1** **Collecting Data**

Source: https: //archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original)

Breast Cancer Diagnostic dataset from the UCI Machine Learning Repository. The breast cancer data includes 569 examples of cancer biopsies, each with 32 features. One feature is an identification number, another is the cancer diagnosis, and 30 are numeric-valued laboratory measurements. The diagnosis is coded as "M" to indicate malignant or "B" to indicate benign.

**Step 2 Exploring and Preparing Data**

Code to import data set

wbcd <- read.csv("wisc\_bc\_data.csv", stringsAsFactors = FALSE)

Code to get structure of data set

str(wbcd)

|  |
| --- |
| 'data.frame': 569 obs. of 32 variables:  $ id : int 87139402 8910251 905520 868871 9012568 906539 925291 87880 862989 89827 ...  $ diagnosis : chr "B" "B" "B" "B" ...  $ radius\_mean : num 12.3 10.6 11 11.3 15.2 ...  $ texture\_mean : num 12.4 18.9 16.8 13.4 13.2 ...  $ perimeter\_mean : num 78.8 69.3 70.9 73 97.7 ...  $ area\_mean : num 464 346 373 385 712 ...  $ smoothness\_mean : num 0.1028 0.0969 0.1077 0.1164 0.0796 ...  $ compactness\_mean : num 0.0698 0.1147 0.078 0.1136 0.0693 ...  $ concavity\_mean : num 0.0399 0.0639 0.0305 0.0464 0.0339 ...  $ points\_mean : num 0.037 0.0264 0.0248 0.048 0.0266 ...  $ symmetry\_mean : num 0.196 0.192 0.171 0.177 0.172 ...  $ dimension\_mean : num 0.0595 0.0649 0.0634 0.0607 0.0554 ...  $ radius\_se : num 0.236 0.451 0.197 0.338 0.178 ...  $ texture\_se : num 0.666 1.197 1.387 1.343 0.412 ...  $ perimeter\_se : num 1.67 3.43 1.34 1.85 1.34 ...  $ area\_se : num 17.4 27.1 13.5 26.3 17.7 ...  $ smoothness\_se : num 0.00805 0.00747 0.00516 0.01127 0.00501 ...  $ compactness\_se : num 0.0118 0.03581 0.00936 0.03498 0.01485 ...  $ concavity\_se : num 0.0168 0.0335 0.0106 0.0219 0.0155 ...  $ points\_se : num 0.01241 0.01365 0.00748 0.01965 0.00915 ...  $ symmetry\_se : num 0.0192 0.035 0.0172 0.0158 0.0165 ...  $ dimension\_se : num 0.00225 0.00332 0.0022 0.00344 0.00177 ...  $ radius\_worst : num 13.5 11.9 12.4 11.9 16.2 ...  $ texture\_worst : num 15.6 22.9 26.4 15.8 15.7 ...  $ perimeter\_worst : num 87 78.3 79.9 76.5 104.5 ...  $ area\_worst : num 549 425 471 434 819 ...  $ smoothness\_worst : num 0.139 0.121 0.137 0.137 0.113 ...  $ compactness\_worst: num 0.127 0.252 0.148 0.182 0.174 ...  $ concavity\_worst : num 0.1242 0.1916 0.1067 0.0867 0.1362 ...  $ points\_worst : num 0.0939 0.0793 0.0743 0.0861 0.0818 ...  $ symmetry\_worst : num 0.283 0.294 0.3 0.21 0.249 ...  $ dimension\_worst : num 0.0677 0.0759 0.0788 0.0678 0.0677 ... |
|  |
| |  | | --- | |  | |

Code to drop id from data set

wbcd <- wbcd[-1]

The table() output indicates that 357 masses are benign while 212 are malignant in diagnosis.

table(wbcd$diagnosis)

B M

357 212

Code to give the "B" and "M" values more informative labels using the labels parameter.

wbcd$diagnosis<- factor(wbcd$diagnosis, levels = c("B", "M"),labels = c("Benign", "Malignant"))

we can check it.

## wbcd$diagnosis

The values have been labeled Benign and Malignant with 62.7 percent and 37.3 percent of the masses.

round(prop.table(table(wbcd$diagnosis)) \* 100, digits = 1)

Benign Malignant

62.7 37.3

We will take summary of 3 character sticks.

summary(wbcd[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

* Transformation -Normalizing Numerical Data

First we will create a function for normalization

normalize <- function(x)

{

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

}

We can check that function is working properly or not.

normalize(c(1, 2, 3, 4, 5))

[1] 0.00 0.25 0.50 0.75 1.00

Code for normalizing data set

wbcd\_n <- as.data.frame(lapply(wbcd[2:31], normalize))

lapply() can be used to apply normalize() to each feature in the data frame.

summary(wbcd\_n$area\_mean)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.0000 0.1174 0.1729 0.2169 0.2711 1.0000

The area mean variable, which originally ranged from 143.5 to 2501.0, now ranges from 0 to 1.

* Data Preparation -Creating training and test datasets

We will use the first 469 records for the training dataset and the remaining 100 to simulate new patients.

# Createing traning data set

wbcd\_train <- wbcd\_n[1:469, ]

# creating test data set

wbcd\_test <- wbcd\_n[470:569, ]

# class labels in factor vectors, split between the training

and test datasets

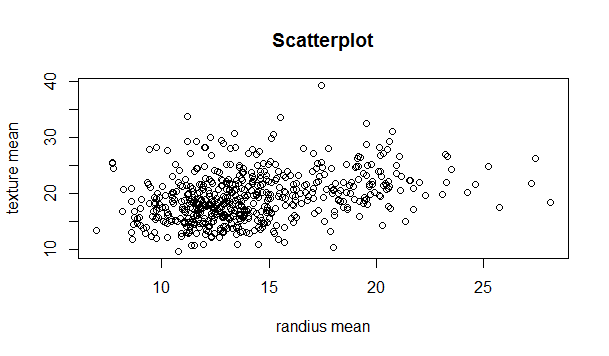
wbcd\_train\_labels <- wbcd[1:469, 1]

wbcd\_test\_labels <- wbcd[470:569, 1]

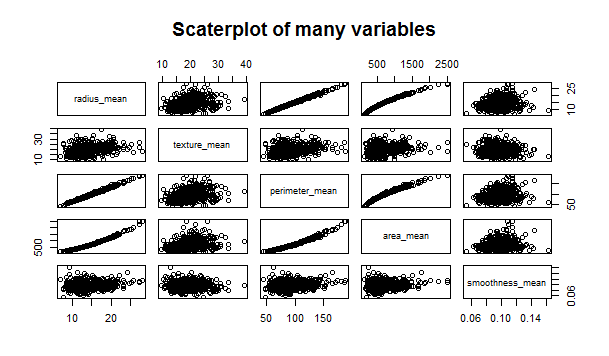
#Visualisation of data using levels

plot(wbcd$radius\_mean,wbcd$texture\_mean, main = 'Scatterplot',xlab

='randius mean', ylab = 'texture mean')



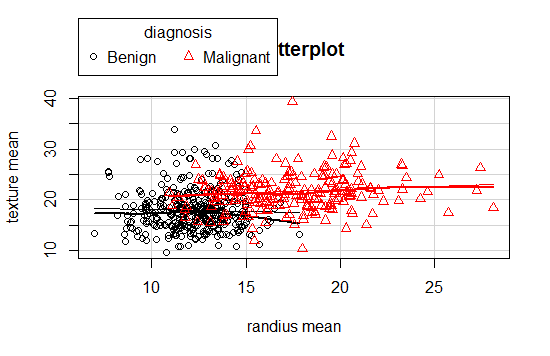
pairs(~radius\_mean+texture\_mean+perimeter\_mean+area\_mean+smoothness\_m ean,data = wbcd, main = 'Scaterplot of many variables')



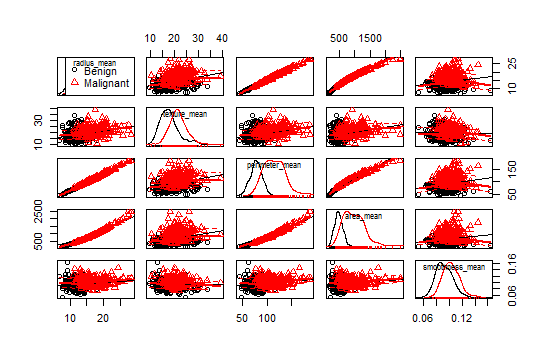
library(car)

scatterplot(texture\_mean ~ radius\_mean | diagnosis, data = wbcd,

main = 'Scatterplot',xlab = 'randius mean',ylab = 'texture mean')



scatterplotMatrix(~radius\_mean+texture\_mean+perimeter\_mean+area\_mean+sm oothness\_mean | diagnosis, data=wbcd)



**Step 3: Training a model on a data**

install.packages("class")

library(class)

wbcd\_test\_pred <- knn(train = wbcd\_train, test = wbcd\_test, cl = wbcd\_train\_labels, k = 3)

The knn() function returns a factor vector of predicted labels for each of the examples in the test dataset, which we have assigned to wbcd\_test\_pred.

wbcd\_test\_pred <- knn(train = wbcd\_train, test = wbcd\_test,

cl = wbcd\_train\_labels, k = 21)

# Head function to get rows of wbcd\_test

head(wbcd\_test)

radius\_mean texture\_mean perimeter\_mean area\_mean smoothness\_mean

470 0.3340906 0.2120392 0.3178080 0.1983881 0.2884355

471 0.2739836 0.3956713 0.2641835 0.1543584 0.3147061

472 0.3781059 0.3398715 0.3573354 0.2318982 0.2850952

473 0.2862890 0.2945553 0.2682607 0.1613150 0.3358310

474 0.5939230 0.7696990 0.5819225 0.4579003 0.2850050

475 0.2394340 0.6232668 0.2284569 0.1299682 0.3149770

compactness\_mean concavity\_mean points\_mean symmetry\_mean dimension\_mean

470 0.12137292 0.08280225 0.14632207 0.3303030 0.18997473

471 0.14302804 0.07291471 0.14234592 0.3202020 0.27190396

472 0.10471750 0.04561856 0.09637177 0.2297980 0.05686605

473 0.05607018 0.06002812 0.14527833 0.2055556 0.18260320

474 0.28716030 0.26827554 0.32987078 0.1858586 0.06676495

475 0.12459358 0.05545923 0.11814115 0.4010101 0.14785173

radius\_se texture\_se perimeter\_se area\_se smoothness\_se compactness\_se

470 0.10056129 0.1515647 0.08891297 0.04844620 0.21290410 0.08034668

471 0.22437081 0.3067097 0.20548462 0.08750126 0.09715471 0.11752336

472 0.02625385 0.1557638 0.02492579 0.01837512 0.06530238 0.04304233

473 0.02621764 0.4379862 0.01946002 0.01374305 0.08971003 0.01988013

474 0.22730400 0.2123409 0.18682561 0.19256329 0.13006085 0.18166251

475 0.17910556 0.5045085 0.15747067 0.07127034 0.19730088 0.09987382

concavity\_se points\_se symmetry\_se dimension\_se radius\_worst texture\_worst

470 0.04060606 0.1713582 0.17121630 0.06685737 0.2749911 0.2547974

471 0.05494949 0.3328282 0.36370800 0.17205616 0.2070438 0.3059701

472 0.02087879 0.1218413 0.15981877 0.02159944 0.2863750 0.3678038

473 0.03391414 0.2204963 0.26492936 0.03047828 0.1910352 0.2875800

474 0.06727273 0.2042053 0.07651826 0.04702680 0.7104233 0.8899254

475 0.03699495 0.2240955 0.17712613 0.10323646 0.2017076 0.5679638

perimeter\_worst area\_worst smoothness\_worst compactness\_worst concavity\_worst

470 0.2529508 0.13881243 0.3191574 0.09499277 0.08921725

471 0.1923901 0.09690818 0.1499703 0.06062811 0.04142173

472 0.2584292 0.14670173 0.2201017 0.07049510 0.06190096

473 0.1695802 0.08865022 0.1706399 0.01833687 0.03860224

474 0.6463967 0.56326190 0.4598164 0.37101610 0.31908946

475 0.1834255 0.09398348 0.2174602 0.06788524 0.04412141

points\_worst symmetry\_worst dimension\_worst

470 0.2561168 0.22866154 0.11347239

471 0.1640206 0.12103292 0.08966286

472 0.1977320 0.21308890 0.03810836

473 0.1722680 0.08318549 0.04361800

474 0.5584192 0.22629608 0.13537977

475 0.1906186 0.16538537 0.07444576

head(wbcd\_test\_pred)

[1] Benign Benign Benign Benign Malignant Benign

Levels: Benign Malignant

**Step 4 :Evaluating model performance**

# load the "gmodels" library

library(gmodels)

#Create the cross tabulation of predicted vs. actual

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 |

| 0.968 | 0.000 | |

| 0.610 | 0.000 | |

-----------------|-----------|-----------|-----------|

Malignant | 2 | 37 | 39 |

| 0.051 | 0.949 | 0.390 |

| 0.032 | 1.000 | |

| 0.020 | 0.370 | |

-----------------|-----------|-----------|-----------|

Column Total | 63 | 37 | 100 |

| 0.630 | 0.370 | |

-----------------|-----------|-----------|-----------|

**Step 5: Improving model performance**

#Create the cross tabulation of predicted vs. actual

wbcd\_z <- as.data.frame(scale(wbcd[-1]))

#Create the cross tabulation of predicted vs. actual

wbcd\_z <- as.data.frame(scale(wbcd[-1]))

# checking that the transformation was applied correctly

summary(wbcd\_z$area\_mean)

Min. 1st Qu. Median Mean 3rd Qu. Max.

-1.4530 -0.6666 -0.2949 0.0000 0.3632 5.2460

# create training and test datasets

wbcd\_train <- wbcd\_z[1:469, ]

wbcd\_test <- wbcd\_z[470:569, ]

wbcd\_test\_pred <- knn(train = wbcd\_train, test = wbcd\_test,cl = wbcd\_train\_labels, k = 21)

# Create the cross tabulation of predicted vs. actual

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 | |

-----------------|-----------|-----------|-----------|

# try several different values of k

wbcd\_train <- wbcd\_n[1:469, ]

wbcd\_test <- wbcd\_n[470:569, ]

wbcd\_test\_pred <- knn(train = wbcd\_train, test = wbcd\_test, cl = wbcd\_train\_labels, k=1)

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 | 58 | 3 | 61 |

| 0.951 | 0.049 | 0.610 |

| 0.983 | 0.073 | |

| 0.580 | 0.030 | |

-----------------|-----------|-----------|-----------|

Malignant | 1 | 38 | 39 |

| 0.026 | 0.974 | 0.390 |

| 0.017 | 0.927 | |

| 0.010 | 0.380 | |

-----------------|-----------|-----------|-----------|

Column Total | 59 | 41 | 100 |

| 0.590 | 0.410 | |

-----------------|-----------|-----------|-----------|

wbcd\_test\_pred <- knn(train = wbcd\_train, test = wbcd\_test, cl = wbcd\_train\_labels, k=5)

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 |

| 0.968 | 0.000 | |

| 0.610 | 0.000 | |

-----------------|-----------|-----------|-----------|

Malignant | 2 | 37 | 39 |

| 0.051 | 0.949 | 0.390 |

| 0.032 | 1.000 | |

| 0.020 | 0.370 | |

-----------------|-----------|-----------|-----------|

Column Total | 63 | 37 | 100 |

| 0.630 | 0.370 | |

-----------------|-----------|-----------|-----------|

wbcd\_test\_pred <- knn(train = wbcd\_train, test = wbcd\_test, cl = wbcd\_train\_labels, k=11)

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 |

| 0.953 | 0.000 | |

| 0.610 | 0.000 | |

-----------------|-----------|-----------|-----------|

Malignant | 3 | 36 | 39 |

| 0.077 | 0.923 | 0.390 |

| 0.047 | 1.000 | |

| 0.030 | 0.360 | |

-----------------|-----------|-----------|-----------|

Column Total | 64 | 36 | 100 |

| 0.640 | 0.360 | |

-----------------|-----------|-----------|-----------|

-----------------|-----------|-----------|-----------|

wbcd\_test\_pred <- knn(train = wbcd\_train, test = wbcd\_test, cl = wbcd\_train\_labels, k=15)

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 |

| 0.953 | 0.000 | |

| 0.610 | 0.000 | |

-----------------|-----------|-----------|-----------|

Malignant | 3 | 36 | 39 |

| 0.077 | 0.923 | 0.390 |

| 0.047 | 1.000 | |

| 0.030 | 0.360 | |

-----------------|-----------|-----------|-----------|

Column Total | 64 | 36 | 100 |

| 0.640 | 0.360 | |

-----------------|-----------|-----------|-----------|

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 |

| 0.968 | 0.000 | |

| 0.610 | 0.000 | |

-----------------|-----------|-----------|-----------|

Malignant | 2 | 37 | 39 |

| 0.051 | 0.949 | 0.390 |

| 0.032 | 1.000 | |

| 0.020 | 0.370 | |

-----------------|-----------|-----------|-----------|

Column Total | 63 | 37 | 100 |

| 0.630 | 0.370 | |

-----------------|-----------|-----------|-----------|

wbcd\_test\_pred <- knn(train = wbcd\_train, test = wbcd\_test, cl = wbcd\_train\_labels, k=27)

CrossTable(x = wbcd\_test\_labels, y = wbcd\_test\_pred, prop.chisq=FALSE)

ell 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 |

| 0.938 | 0.000 | |

| 0.610 | 0.000 | |

-----------------|-----------|-----------|-----------|

Malignant | 4 | 35 | 39 |

| 0.103 | 0.897 | 0.390 |

| 0.062 | 1.000 | |

| 0.040 | 0.350 | |

-----------------|-----------|-----------|-----------|

Column Total | 65 | 35 | 100 |

| 0.650 | 0.350 | |

-----------------|-----------|-----------|-----------|

performing knn for 1,5,11,15,21,27.It is clear that the kvalue of 5 and 21 has more accuracy.

|  |  |  |  |
| --- | --- | --- | --- |
| K-Value | False Negative | False Positive | Accuracy |
| K=1 | 1 | 3 | 96% |
| K=5 | 2 | 0 | 98% |
| K=11 | 3 | 0 | 97% |
| K=15 | 3 | 0 | 97% |
| K=21 | 2 | 0 | 98% |
| K=27 | 4 | 0 | 96% |

|  |
| --- |
| **Bank Note Authentication Data Set** |

**Step 1:**

Source : <https://archive.ics.uci.edu/ml/machine-learning-databases/00267/>

Data information: Data were extracted from images that were taken from genuine and forged banknote-like specimens. For digitization, an industrial camera usually used for print inspection was used. The final images have 400x 400 pixels. Due to the object lens and distance to the investigated object gray-scale pictures with a resolution of about 660 dpi were gained. Wavelet Transform tool were used to extract features from images.

#Read Data

Banknote=read.table("C:/Users/chink/Desktop/data\_banknote\_authentication.txt",sep=",",header = F)

#Applying headers

colnames(Banknote)[1]="variance\_Wavelet"

colnames(Banknote)[2]="skewness\_Wavelet"

colnames(Banknote)[3]="curtosis\_Wavelet"

colnames(Banknote)[4]="entropy"

colnames(Banknote)[5]="class"

View(Banknote)

# The dataset is too huge for knn analysis, we have chosen random 20%

divide=sample(nrow(Banknote),floor(nrow(Banknote)\*0.2))

Banknote=Banknote[divide,]

Step 2: Data Exploration and Preparation

#Examine structure of data frame

str(Banknote)

'data.frame': 274 obs. of 5 variables:

$ variance\_Wavelet: num 3.19 0.662 -2.234 -2.292 4.095 ...

$ skewness\_Wavelet: num 5.75 9.66 -7.03 -7.26 -2.97 ...

$ curtosis\_Wavelet: num -0.185 -0.288 7.494 7.96 2.369 ...

$ entropy : num -0.301 -1.664 0.613 0.921 0.754 ...

$ class : int 0 0 1 1 0 0 1 0 0 1 ...

# Record class as a factor

Banknote$class=factor(Banknote$class,levels = c(0,1),labels=c("Fake","correct"))

#Table of class

table(Banknote$class)

Fake correct

150 124

#Proportion with more informative

round(prop.table(table(Banknote$class))\*100, digits = 2)

Fake correct

54.74 45.26

#summary of three variables

summary(Banknote[c("variance\_Wavelet","skewness\_Wavelet","curtosis\_Wavelet","entropy")])

variance\_Wavelet skewness\_Wavelet curtosis\_Wavelet entropy

Min. :-7.0421 Min. :-13.459 Min. :-4.7275 Min. :-7.8719

1st Qu.:-1.8548 1st Qu.: -1.826 1st Qu.:-1.2750 1st Qu.:-2.4630

Median : 0.2493 Median : 2.357 Median : 0.6873 Median :-0.5291

Mean : 0.2886 Mean : 1.965 Mean : 1.3861 Mean :-1.1294

3rd Qu.: 3.1315 3rd Qu.: 6.754 3rd Qu.: 3.1268 3rd Qu.: 0.4747

Max. : 5.9374 Max. : 12.118 Max. :17.5932 Max. : 1.9284

# create normalization function

normalize <- function(x) {

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

}

# test normalization function - result should be identical

normalize(c(1, 2, 3, 4, 5))

[1] 0.00 0.25 0.50 0.75 1.00

normalize(c(10, 20, 30, 40, 50))

[1] 0.00 0.25 0.50 0.75 1.00

#Normalized Data Set

Banknote\_normalized=as.data.frame(lapply(Banknote[1:4],normalize))

#confirm that normalization is working

summary(Banknote\_normalized$skewness\_Wavelet)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.0000 0.4548 0.6184 0.6030 0.7903 1.0000

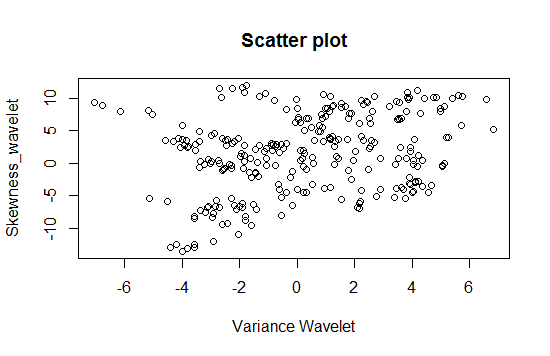
# create training and test data

Banknote\_training=Banknote\_normalized[1:200,]

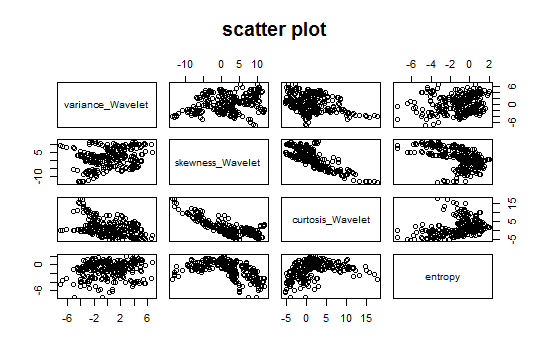
Banknote\_test=Banknote\_normalized[201:274,]

# visualize the data using labels

plot(Banknote$variance\_Wavelet,Banknote$skewness\_Wavelet,main="Scatter plot",xlab = "Variance Wavelet",ylab="Skewness\_wavelet")



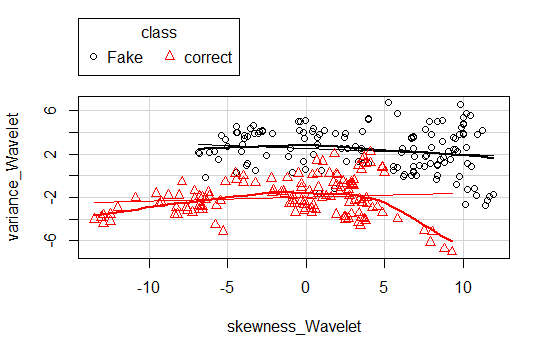
pairs(~variance\_Wavelet+skewness\_Wavelet+curtosis\_Wavelet+entropy,data = Banknote,main="scatter plot")



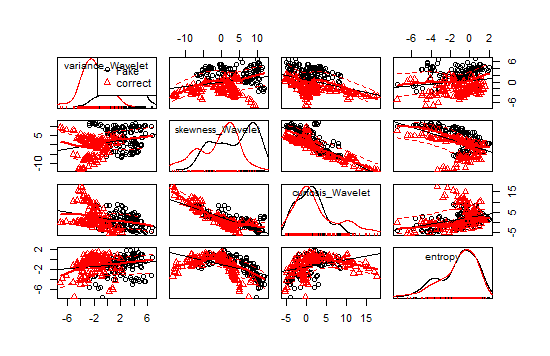
#Uploading library car

library(car)

scatterplot(variance\_Wavelet ~ skewness\_Wavelet |class , data = Banknote)



scatterplotMatrix(~variance\_Wavelet+skewness\_Wavelet+curtosis\_Wavelet+entropy | class, data = Banknote)



Step 3: Training a model on Data

# load the "class" library

library(class)

Banknote\_test\_pred=knn(train = Banknote\_training, test = Banknote\_test, cl=Banknote\_training\_labels,k=3)

head(Banknote\_test)

variance\_Wavelet skewness\_Wavelet curtosis\_Wavelet entropy

201 0.3885974 0.2678808 0.42798389 0.7603910

202 0.8325128 0.6349511 0.24142119 0.8381060

203 0.3816018 0.5072814 0.28814060 0.9811332

204 0.4952178 0.5226535 0.19754398 0.8544677

205 0.8141916 0.5763734 0.29815821 0.9868065

206 0.6115613 0.8302569 0.08871586 0.6057570

head(Banknote\_test\_pred)

[1] correct Fake correct correct Fake Fake

Levels: Fake correct

summary(Banknote\_test\_pred)

Fake correct

39 35

**Step 4: Evaluating model performance**

# load the "gmodels" library

library(gmodels)

# Create the cross tabulation of predicted vs. actual

CrossTable(x=Banknote\_test\_labels ,y=Banknote\_test\_pred ,prop.chisq = FALSE)

Cell Contents

|-------------------------|

| N |

| N / Row Total |

| N / Col Total |

| N / Table Total |

|-------------------------|

Total Observations in Table: 74

| Banknote\_test\_pred

Banknote\_test\_labels | Fake | correct | Row Total |

---------------------|-----------|-----------|-----------|

Fake | 39 | 0 | 39 |

| 1.000 | 0.000 | 0.527 |

| 1.000 | 0.000 | |

| 0.527 | 0.000 | |

---------------------|-----------|-----------|-----------|

correct | 0 | 35 | 35 |

| 0.000 | 1.000 | 0.473 |

| 0.000 | 1.000 | |

| 0.000 | 0.473 | |

---------------------|-----------|-----------|-----------|

Column Total | 39 | 35 | 74 |

| 0.527 | 0.473 | |

---------------------|-----------|-----------|-----------|

**Step:5 Improving model performance**

# use the scale() function to z-score standardize a data frame

Banknote\_z=as.data.frame(scale(Banknote[-5]))

# confirm that the transformation was applied correctly

summary(Banknote$variance\_Wavelet)

Min. 1st Qu. Median Mean 3rd Qu. Max.

-7.0420 -1.8550 0.2494 0.2886 3.1310 5.9370

# create training and test datasets

Banknote\_training=Banknote\_z[1:200,]

Banknote\_test=Banknote\_z[201:274,]

# re-classify test cases

Banknote\_test\_pred=knn(train = Banknote\_training, test = Banknote\_test, cl=Banknote\_training\_labels,k=3)

CrossTable(x=Banknote\_test\_labels,y=Banknote\_test\_pred,prop.chisq = FALSE)

Cell Contents

|-------------------------|

| N |

| N / Row Total |

| N / Col Total |

| N / Table Total |

|-------------------------|

Total Observations in Table: 74

| Banknote\_test\_pred

Banknote\_test\_labels | Fake | correct | Row Total |

---------------------|-----------|-----------|-----------|

Fake | 39 | 0 | 39 |

| 1.000 | 0.000 | 0.527 |

| 1.000 | 0.000 | |

| 0.527 | 0.000 | |

---------------------|-----------|-----------|-----------|

correct | 0 | 35 | 35 |

| 0.000 | 1.000 | 0.473 |

| 0.000 | 1.000 | |

| 0.000 | 0.473 | |

---------------------|-----------|-----------|-----------|

Column Total | 39 | 35 | 74 |

| 0.527 | 0.473 | |

---------------------|-----------|-----------|-----------|

# try several different values of k

Banknote\_training=Banknote\_normalized[1:200,]

Banknote\_test=Banknote\_normalized[201:274,]

Banknote\_test\_pred=knn(train = Banknote\_training, test = Banknote\_test, cl=Banknote\_training\_labels,k=11)

CrossTable(x=Banknote\_test\_labels,y=Banknote\_test\_pred,prop.chisq = FALSE)

Cell Contents

|-------------------------|

| N |

| N / Row Total |

| N / Col Total |

| N / Table Total |

|-------------------------|

Total Observations in Table: 74

| Banknote\_test\_pred

Banknote\_test\_labels | Fake | correct | Row Total |

---------------------|-----------|-----------|-----------|

Fake | 38 | 1 | 39 |

| 0.974 | 0.026 | 0.527 |

| 1.000 | 0.028 | |

| 0.514 | 0.014 | |

---------------------|-----------|-----------|-----------|

correct | 0 | 35 | 35 |

| 0.000 | 1.000 | 0.473 |

| 0.000 | 0.972 | |

| 0.000 | 0.473 | |

---------------------|-----------|-----------|-----------|

Column Total | 38 | 36 | 74 |

| 0.514 | 0.486 | |

---------------------|-----------|-----------|-----------|

Banknote\_test\_pred=knn(train = Banknote\_training, test = Banknote\_test, cl=Banknote\_training\_labels,k=13)

CrossTable(x=Banknote\_test\_labels,y=Banknote\_test\_pred,prop.chisq = FALSE)

Cell Contents

|-------------------------|

| N |

| N / Row Total |

| N / Col Total |

| N / Table Total |

|-------------------------|

Total Observations in Table: 74

| Banknote\_test\_pred

Banknote\_test\_labels | Fake | correct | Row Total |

---------------------|-----------|-----------|-----------|

Fake | 38 | 1 | 39 |

| 0.974 | 0.026 | 0.527 |

| 1.000 | 0.028 | |

| 0.514 | 0.014 | |

---------------------|-----------|-----------|-----------|

correct | 0 | 35 | 35 |

| 0.000 | 1.000 | 0.473 |

| 0.000 | 0.972 | |

| 0.000 | 0.473 | |

---------------------|-----------|-----------|-----------|

Column Total | 38 | 36 | 74 |

| 0.514 | 0.486 | |

---------------------|-----------|-----------|-----------|

Banknote\_test\_pred=knn(train = Banknote\_training, test = Banknote\_test, cl=Banknote\_training\_labels,k=15)

CrossTable(x=Banknote\_test\_labels,y=Banknote\_test\_pred,prop.chisq = FALSE)

Cell Contents

|-------------------------|

| N |

| N / Row Total |

| N / Col Total |

| N / Table Total |

|-------------------------|

Total Observations in Table: 74

| Banknote\_test\_pred

Banknote\_test\_labels | Fake | correct | Row Total |

---------------------|-----------|-----------|-----------|

Fake | 38 | 1 | 39 |

| 0.974 | 0.026 | 0.527 |

| 1.000 | 0.028 | |

| 0.514 | 0.014 | |

---------------------|-----------|-----------|-----------|

correct | 0 | 35 | 35 |

| 0.000 | 1.000 | 0.473 |

| 0.000 | 0.972 | |

| 0.000 | 0.473 | |

---------------------|-----------|-----------|-----------|

Column Total | 38 | 36 | 74 |

| 0.514 | 0.486 | |

---------------------|-----------|-----------|-----------|

Banknote\_test\_pred=knn(train = Banknote\_training, test = Banknote\_test, cl=Banknote\_training\_labels,k=5)

CrossTable(x=Banknote\_test\_labels,y=Banknote\_test\_pred,prop.chisq = FALSE)

Cell Contents

|-------------------------|

| N |

| N / Row Total |

| N / Col Total |

| N / Table Total |

|-------------------------|

Total Observations in Table: 74

| Banknote\_test\_pred

Banknote\_test\_labels | Fake | correct | Row Total |

---------------------|-----------|-----------|-----------|

Fake | 39 | 0 | 39 |

| 1.000 | 0.000 | 0.527 |

| 1.000 | 0.000 | |

| 0.527 | 0.000 | |

---------------------|-----------|-----------|-----------|

correct | 0 | 35 | 35 |

| 0.000 | 1.000 | 0.473 |

| 0.000 | 1.000 | |

| 0.000 | 0.473 | |

---------------------|-----------|-----------|-----------|

Column Total | 39 | 35 | 74 |

| 0.527 | 0.473 | |

---------------------|-----------|-----------|-----------|

Banknote\_test\_pred=knn(train = Banknote\_training, test = Banknote\_test, cl=Banknote\_training\_labels,k=23)

CrossTable(x=Banknote\_test\_labels,y=Banknote\_test\_pred,prop.chisq = FALSE)

Cell Contents

|-------------------------|

| N |

| N / Row Total |

| N / Col Total |

| N / Table Total |

|-------------------------|

Total Observations in Table: 74

| Banknote\_test\_pred

Banknote\_test\_labels | Fake | correct | Row Total |

---------------------|-----------|-----------|-----------|

Fake | 38 | 1 | 39 |

| 0.974 | 0.026 | 0.527 |

| 0.950 | 0.029 | |

| 0.514 | 0.014 | |

---------------------|-----------|-----------|-----------|

correct | 2 | 33 | 35 |

| 0.057 | 0.943 | 0.473 |

| 0.050 | 0.971 | |

| 0.027 | 0.446 | |

---------------------|-----------|-----------|-----------|

Column Total | 40 | 34 | 74 |

| 0.541 | 0.459 | |

---------------------|-----------|-----------|-----------|

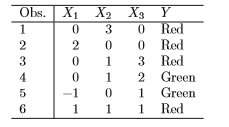
Performing KNN for 1,5,11,15 and 23. It is clear k=1,5 yields more

accuracy.

|  |  |  |  |
| --- | --- | --- | --- |
| K Value | False Negative | False Positive | Accuracy |
| K=1 | 0 | 0 | 100% |
| K=5 | 0 | 0 | 100% |
| K=11 | 0 | 1 | 99% |
| K=15 | 0 | 1 | 99% |
| K=23 | 2 | 1 | 97% |

**Problem 7:**

The table below provides a training data set containing six observations, three predictors, and one qualitative response variable.



Suppose we wish to use this data set to make a prediction for Y when X1 = X2 = X3 = 0 using K-nearest neighbors.

1. Compute the Euclidean distance between each observation and the test point, X1 = X2 = X3 = 0.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Obs. | X1 | X2 | X3 | Y | X5 |
| 1 | 0 | 3 | 0 | Red | 3 |
| 2 | 2 | 0 | 0 | Red | 2 |
| 3 | 0 | 1 | 3 | Red | 3.162278 |
| 4 | 0 | 1 | 2 | Green | 2.236068 |
| 5 | -1 | 0 | 1 | Green | 1.414214 |
| 6 | 1 | 1 | 1 | Red | 1.732051 |

R Code:

x1=c(0,3,0)

x2=c(2,0,0)

x3=c(0,1,3)

x4=c(0,1,2)

x5=c(-1,0,1)

x6=c(1,1,1)

A=c(0,0,0)

dist(rbind(x1,A))

dist(rbind(x2,A))

dist(rbind(x3,A))

dist(rbind(x4,A))

dist(rbind(x5,A))

dist(rbind(x6,A))

1. What is our prediction with K = 1? Why?

With the k=1 , The nearest test point of (0,0,0) is observation 5 because distance between point(0,0,0) and x5 is 1.414214.

Green would be prediction.

1. What is our prediction with K = 3? Why?

With the k=2, The nearest test point of (0,0,0) is observation 5,6 and 2 because distance between point(0,0,0) and observation X5, X6 ,X2 is min.

Red would be prediction because red occurs twice.

1. If the Bayes decision boundary in this problem is highly nonlinear, then would we expect

the best value for K to be large or small? Why

If the Bayes decision boundary is highly nonlinear then k value will be small. If K value is large then KNN decision boundary would be linear and not sufficiently flexible and If k value is small then KNN decision boundary would be nonlinear which will be Bayes decision boundary.