**Lab 13: k-Nearest Neighbor Classifier**

**Problem statement:**

Task1: Refer attached dataset wisc\_bc\_data.csv (Cancer dataset) as

* Total number of columns: 32
* Column-1 PatientID -- should be discarded
* Column-2- class label (Two classes ‘B’ and ‘M’)
* Remaining columns are numerical values representing values against various test
* Convert column-2 (diagnosis) as factor
* Normalized all numeric feature and scale values between 0 to 1
* Create test and train dataset as per your choice
* Develop KNN classification with k=21
* Train KNN model with train dataset
* Test the KNN model with test dataset
* Observe the confusion matrix
* Repeat the same procedure with K=1 to K= number of training dataset and observe the classifier performance

Task-2: Repeat this with different scaling formula (Z-score standardization)

* X\_new= (x-Mean(x))/ StdDev (x)

**Source Code:**

#Author: Ashish Upadhyay

#Branch: Computer Science and Engineering

#Semester: 6th

#Dr. SP Mukherjee International Institute of Information Technology, Naya Raipur

#Subject: Machine Learning Lab 13

#Task: k-Nearest Neighbour Implementation

setwd("C:/Users/Ashish Upadhyay/Documents/Semester6/MachineLearning/Lab Programs")

getwd()

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

stringsAsFactors = FALSE

#head(data\_set)

nrow(data\_set)

str(data\_set)

data\_set <- data\_set[-1]

str(data\_set)

table(data\_set$diagnosis)

data\_set$diagnosis <- as.factor(data\_set$diagnosis)

#Normalization

normalize <- function(x) {

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

}

# One could also use sequence such as df[1:2]

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

head(dfNorm)

data\_train <- dfNorm[1:400,]

data\_test <- dfNorm[401:569,]

data\_train\_labels <- data\_set[1:400, 1]

data\_test\_labels <- data\_set[401:569, 1]

#install.packages("class")

library(class)

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

#summary(data\_test\_pred)

#install.packages("gmodels")

library(gmodels)

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

#Z-score standardization

stad <- function(x) {

return ((x - mean(x)) / sqrt(var(x)))

}

# One could also use sequence such as df[1:2]

dfNorm <- as.data.frame(lapply(data\_set[2:31], stad))

head(dfNorm)

data\_train <- dfNorm[1:400,]

data\_test <- dfNorm[401:569,]

data\_train\_labels <- data\_set[1:400, 1]

data\_test\_labels <- data\_set[401:569, 1]

#install.packages("class")

library(class)

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

#summary(data\_test\_pred)

#install.packages("gmodels")

library(gmodels)

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

**Output:**

> #Author: Ashish Upadhyay

> #Branch: Computer Science and Engineering

> #Semester: 6th

> #Dr. SP Mukherjee International Institute of Information Technology, Naya Raipur

> #Subject: Machine Learning Lab 13

> #Task: k-Nearest Neighbour Implementation

>

> setwd("C:/Users/Ashish Upadhyay/Documents/Semester6/MachineLearning/Lab Programs")

> getwd()

[1] "C:/Users/Ashish Upadhyay/Documents/Semester6/MachineLearning/Lab Programs"

>

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

> stringsAsFactors = FALSE

> #head(data\_set)

> nrow(data\_set)

[1] 569

> str(data\_set)

'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 ...

> data\_set <- data\_set[-1]

> str(data\_set)

'data.frame': 569 obs. of 31 variables:

$ 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 ...

> table(data\_set$diagnosis)

B M

357 212

> data\_set$diagnosis <- as.factor(data\_set$diagnosis)

>

> #Normalization

> normalize <- function(x) {

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

+ }

> # One could also use sequence such as df[1:2]

> dfNorm <- as.data.frame(lapply(data\_set[2:31], normalize))

> head(dfNorm)

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

1 0.2526859 0.0906324 0.2422777 0.13599152 0.4529205

2 0.1712812 0.3124789 0.1761454 0.08606575 0.3994764

3 0.1921056 0.2407846 0.1874784 0.09743372 0.4971563

4 0.2034644 0.1244505 0.2018520 0.10235419 0.5756974

5 0.3885182 0.1183632 0.3721927 0.24106045 0.2437483

6 0.2171896 0.3155225 0.2101444 0.11291622 0.2963799

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

1 0.1546838 0.09341612 0.18389662 0.4540404 0.2019798

2 0.2923747 0.14964855 0.13131213 0.4353535 0.3148694

3 0.1799276 0.07136832 0.12326044 0.3303030 0.2830666

4 0.2890007 0.10859888 0.23836978 0.3590909 0.2266217

5 0.1532421 0.07949859 0.13205765 0.3338384 0.1154170

6 0.1774124 0.12851453 0.07097416 0.4904040 0.2676917

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

1 0.04508419 0.06749470 0.04301937 0.01985065 0.2152497 0.07170968

2 0.12275937 0.18493635 0.12594826 0.03791198 0.1957032 0.25203533

3 0.03085280 0.22692716 0.02756443 0.01258503 0.1171092 0.05334665

4 0.08216549 0.21720297 0.05154785 0.03647380 0.3248802 0.24580166

5 0.02418975 0.01155852 0.02737596 0.02039231 0.1121460 0.09461652

6 0.06333514 0.23864038 0.06827498 0.02521115 0.1891763 0.13682519

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

1 0.04250000 0.2350824 0.1598188 0.04675041 0.1981501 0.09648188

2 0.08469697 0.2585717 0.3821410 0.08371682 0.1405194 0.29104478

3 0.02666667 0.1417503 0.1308324 0.04502301 0.1593739 0.38432836

4 0.05522727 0.3722296 0.1114144 0.08800077 0.1419424 0.09994670

5 0.03916667 0.1734230 0.1208420 0.03013280 0.2942014 0.09888060

6 0.11229798 0.1666793 0.1519390 0.08444233 0.1828531 0.39872068

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

1 0.1820808 0.08943669 0.4446279 0.09635106 0.09920128

2 0.1388017 0.05888714 0.3310440 0.21752966 0.15303514

3 0.1470193 0.07034015 0.4340619 0.11730749 0.08522364

4 0.1300862 0.06114825 0.4327412 0.15029446 0.06924121

5 0.2693859 0.15579532 0.2735918 0.14204771 0.10878594

6 0.1793914 0.08240759 0.3548174 0.16145181 0.20447284

points\_worst symmetry\_worst dimension\_worst

1 0.3227148 0.2487680 0.08310376

2 0.2723711 0.2710428 0.13662600

3 0.2553608 0.2824759 0.15590975

4 0.2959107 0.1058545 0.08395645

5 0.2810309 0.1817465 0.08277581

6 0.2290034 0.2897694 0.18234291

>

> data\_train <- dfNorm[1:400,]

> data\_test <- dfNorm[401:569,]

>

> data\_train\_labels <- data\_set[1:400, 1]

> data\_test\_labels <- data\_set[401:569, 1]

>

> #install.packages("class")

> library(class)

>

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

> #summary(data\_test\_pred)

>

> #install.packages("gmodels")

> library(gmodels)

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

Cell Contents

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

| N |

| N / Row Total |

| N / Col Total |

| N / Table Total |

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

Total Observations in Table: 169

| data\_test\_pred

data\_test\_labels | B | M | Row Total |

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

B | 106 | 0 | 106 |

| 1.000 | 0.000 | 0.627 |

| 0.972 | 0.000 | |

| 0.627 | 0.000 | |

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

M | 3 | 60 | 63 |

| 0.048 | 0.952 | 0.373 |

| 0.028 | 1.000 | |

| 0.018 | 0.355 | |

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

Column Total | 109 | 60 | 169 |

| 0.645 | 0.355 | |

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

>

> #Z-score standardization

> stad <- function(x) {

+ return ((x - mean(x)) / sqrt(var(x)))

+ }

> # One could also use sequence such as df[1:2]

> dfNorm <- as.data.frame(lapply(data\_set[2:31], stad))

> head(dfNorm)

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

1 -0.5128453 -1.60418301 -0.5399006 -0.5421468 0.4578825

2 -1.0009202 -0.07896900 -0.9337442 -0.8766033 0.0369535

3 -0.8760638 -0.57187353 -0.8662517 -0.8004484 0.8062867

4 -0.8079604 -1.37168088 -0.7806514 -0.7674858 1.4248817

5 0.3015589 -1.41353126 0.2337944 0.1617181 -1.1895712

6 -0.7256686 -0.05804381 -0.7312666 -0.6967299 -0.7750414

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

1 -0.6538379 -0.6137661 -0.30717196 0.5376080 -0.45997776

2 0.1961461 -0.3127117 -0.57983238 0.4026419 0.29919003

3 -0.4980044 -0.7318045 -0.62158190 -0.3560868 0.08532000

4 0.1753178 -0.5324814 -0.02471844 -0.1481659 -0.29426389

5 -0.6627373 -0.6882771 -0.57596668 -0.3305526 -1.04210082

6 -0.5135309 -0.4258580 -0.89269604 0.8002448 -0.01807412

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

1 -0.6100407 -0.99928403 -0.5915654 -0.5035518 0.33439304 -0.7637928

2 0.1634542 -0.03598928 0.2789225 -0.2909823 0.14288710 0.5769353

3 -0.7517580 0.30843301 -0.7537927 -0.5890632 -0.62713327 -0.9003226

4 -0.2407825 0.22867206 -0.5020436 -0.3079088 1.40849153 0.5305878

5 -0.8181090 -1.45809077 -0.7557711 -0.4971769 -0.67575913 -0.5934796

6 -0.4282964 0.40450870 -0.3264623 -0.4404624 0.07894077 -0.2796565

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

1 -0.49902890 0.09948696 -0.1575418 -0.5846041 -0.5729467 -1.6330626

2 0.05453788 0.30045012 1.7538168 -0.1802309 -0.9081255 -0.4453479

3 -0.70674066 -0.69901746 -0.4067442 -0.6035000 -0.7984682 0.1241043

4 -0.33206441 1.27285250 -0.5736857 -0.1333690 -0.8998495 -1.6119115

5 -0.54275769 -0.42804133 -0.4926344 -0.7663830 -0.0143154 -1.6184195

6 0.41662554 -0.48573721 -0.2252861 -0.1722946 -0.6619139 0.2119626

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

1 -0.60385945 -0.5822061 0.2685394 -0.81141409 -0.70935407

2 -0.86247083 -0.8005226 -0.4847751 -0.01757408 -0.38628525

3 -0.81336741 -0.7186759 0.1984636 -0.67412871 -0.79323693

4 -0.91455023 -0.7843640 0.1897041 -0.45803135 -0.88915098

5 -0.08217273 -0.1079870 -0.8658121 -0.51205569 -0.65183440

6 -0.61992966 -0.6324382 -0.3271047 -0.38493959 -0.07759635

points\_worst symmetry\_worst dimension\_worst

1 -0.3148560 -0.11921566 -0.89893012

2 -0.5377296 0.06343283 -0.44713458

3 -0.6130350 0.15718162 -0.28435531

4 -0.4335191 -1.29107549 -0.89173240

5 -0.4993923 -0.66877751 -0.90169847

6 -0.7297203 0.21698688 -0.06122589

>

> data\_train <- dfNorm[1:400,]

> data\_test <- dfNorm[401:569,]

>

> data\_train\_labels <- data\_set[1:400, 1]

> data\_test\_labels <- data\_set[401:569, 1]

>

> #install.packages("class")

> library(class)

>

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

> #summary(data\_test\_pred)

>

> #install.packages("gmodels")

> library(gmodels)

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

Cell Contents

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

| N |

| N / Row Total |

| N / Col Total |

| N / Table Total |

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

Total Observations in Table: 169

| data\_test\_pred

data\_test\_labels | B | M | Row Total |

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

B | 106 | 0 | 106 |

| 1.000 | 0.000 | 0.627 |

| 0.938 | 0.000 | |

| 0.627 | 0.000 | |

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

M | 7 | 56 | 63 |

| 0.111 | 0.889 | 0.373 |

| 0.062 | 1.000 | |

| 0.041 | 0.331 | |

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

Column Total | 113 | 56 | 169 |

| 0.669 | 0.331 | |

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