HW\_kNN

경제학과 2020110210 공소연

2022-10-21

## Q1

library(mlbench)  
data("PimaIndiansDiabetes2")  
data <- PimaIndiansDiabetes2  
str(data)

## 'data.frame': 768 obs. of 9 variables:  
## $ pregnant: num 6 1 8 1 0 5 3 10 2 8 ...  
## $ glucose : num 148 85 183 89 137 116 78 115 197 125 ...  
## $ pressure: num 72 66 64 66 40 74 50 NA 70 96 ...  
## $ triceps : num 35 29 NA 23 35 NA 32 NA 45 NA ...  
## $ insulin : num NA NA NA 94 168 NA 88 NA 543 NA ...  
## $ mass : num 33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 NA ...  
## $ pedigree: num 0.627 0.351 0.672 0.167 2.288 ...  
## $ age : num 50 31 32 21 33 30 26 29 53 54 ...  
## $ diabetes: Factor w/ 2 levels "neg","pos": 2 1 2 1 2 1 2 1 2 2 ...

summary(data)

## pregnant glucose pressure triceps   
## Min. : 0.000 Min. : 44.0 Min. : 24.00 Min. : 7.00   
## 1st Qu.: 1.000 1st Qu.: 99.0 1st Qu.: 64.00 1st Qu.:22.00   
## Median : 3.000 Median :117.0 Median : 72.00 Median :29.00   
## Mean : 3.845 Mean :121.7 Mean : 72.41 Mean :29.15   
## 3rd Qu.: 6.000 3rd Qu.:141.0 3rd Qu.: 80.00 3rd Qu.:36.00   
## Max. :17.000 Max. :199.0 Max. :122.00 Max. :99.00   
## NA's :5 NA's :35 NA's :227   
## insulin mass pedigree age diabetes   
## Min. : 14.00 Min. :18.20 Min. :0.0780 Min. :21.00 neg:500   
## 1st Qu.: 76.25 1st Qu.:27.50 1st Qu.:0.2437 1st Qu.:24.00 pos:268   
## Median :125.00 Median :32.30 Median :0.3725 Median :29.00   
## Mean :155.55 Mean :32.46 Mean :0.4719 Mean :33.24   
## 3rd Qu.:190.00 3rd Qu.:36.60 3rd Qu.:0.6262 3rd Qu.:41.00   
## Max. :846.00 Max. :67.10 Max. :2.4200 Max. :81.00   
## NA's :374 NA's :11

# 1  
data$glucose[is.na(data$glucose)] <- median(data$glucose, na.rm = T)  
data$pressure[is.na(data$pressure)] <- median(data$pressure, na.rm = T)  
data$triceps[is.na(data$triceps)] <- median(data$triceps, na.rm = T)  
data$insulin[is.na(data$insulin)] <- median(data$insulin, na.rm = T)  
data$mass[is.na(data$mass)] <- median(data$mass, na.rm = T)  
summary(data)

## pregnant glucose pressure triceps   
## Min. : 0.000 Min. : 44.00 Min. : 24.00 Min. : 7.00   
## 1st Qu.: 1.000 1st Qu.: 99.75 1st Qu.: 64.00 1st Qu.:25.00   
## Median : 3.000 Median :117.00 Median : 72.00 Median :29.00   
## Mean : 3.845 Mean :121.66 Mean : 72.39 Mean :29.11   
## 3rd Qu.: 6.000 3rd Qu.:140.25 3rd Qu.: 80.00 3rd Qu.:32.00   
## Max. :17.000 Max. :199.00 Max. :122.00 Max. :99.00   
## insulin mass pedigree age diabetes   
## Min. : 14.0 Min. :18.20 Min. :0.0780 Min. :21.00 neg:500   
## 1st Qu.:121.5 1st Qu.:27.50 1st Qu.:0.2437 1st Qu.:24.00 pos:268   
## Median :125.0 Median :32.30 Median :0.3725 Median :29.00   
## Mean :140.7 Mean :32.46 Mean :0.4719 Mean :33.24   
## 3rd Qu.:127.2 3rd Qu.:36.60 3rd Qu.:0.6262 3rd Qu.:41.00   
## Max. :846.0 Max. :67.10 Max. :2.4200 Max. :81.00

## Q2

set.seed(1)  
train.index <- sample(row.names(data), 0.6\*dim(data)[1])  
valid.index <- sample(setdiff(row.names(data), train.index), 0.2\*dim(data)[1])  
test.index <- setdiff(row.names(data), union(train.index,valid.index))  
  
train.df <- data[train.index,]  
valid.df <- data[valid.index,]  
test.df <- data[test.index,]  
  
train.norm.df <- train.df  
valid.norm.df <- valid.df  
test.norm.df <- test.df  
data.norm <- data  
  
library(caret)

## 필요한 패키지를 로딩중입니다: ggplot2

## 필요한 패키지를 로딩중입니다: lattice

norm.values <- preProcess(train.df[,1:8], method=c("center","scale"))  
  
train.norm.df[, 1:8] <- predict(norm.values, train.df[,1:8])  
valid.norm.df[, 1:8] <- predict(norm.values, valid.df[,1:8])  
test.norm.df[, 1:8] <- predict(norm.values, test.df[,1:8])  
data.norm[, 1:8] <- predict(norm.values, data[,1:8])  
  
head(train.norm.df)

## pregnant glucose pressure triceps insulin mass  
## 679 -0.2720092 -0.03316428 -1.716050417 -0.05159703 -0.17804684 0.4817373  
## 129 -0.8472930 -0.16411337 1.370827152 -0.60874805 0.06819954 0.2682917  
## 509 -0.5596511 -1.24444334 -1.887543615 -0.72017826 -0.78135047 -0.3151260  
## 471 -0.8472930 0.71979297 0.856347557 1.17413523 -0.17804684 1.2359114  
## 299 2.8920518 -0.72064699 0.513361161 -0.49731785 0.54837998 0.5671155  
## 270 -0.5596511 0.78526752 -0.001118434 -0.05159703 -0.17804684 -0.7277874  
## pedigree age diabetes  
## 679 -1.0731712 -0.7195159 pos  
## 129 -0.2600647 0.5266336 pos  
## 509 1.4044468 -1.0518224 neg  
## 471 0.3409271 -0.4702860 neg  
## 299 -0.2335504 1.0250934 pos  
## 270 -0.7402689 -0.4702860 pos

head(valid.norm.df)

## pregnant glucose pressure triceps insulin mass  
## 581 -1.1349349 0.94895388 1.5423204 1.84271646 -0.1780468 1.34974906  
## 306 -0.5596511 -0.06590155 0.3418680 0.83984461 -0.4242932 1.00823622  
## 670 1.4538422 1.04716569 0.5133612 0.05983318 -0.4858548 -0.24397751  
## 283 0.8785584 0.35968298 1.3708272 -1.61161990 0.1913227 -0.03053199  
## 385 -0.8472930 0.09778481 -0.1726116 -0.60874805 -0.3627316 -1.18313781  
## 589 -0.2720092 1.76738567 1.1993340 -0.27445744 0.2036351 0.09753532  
## pedigree age diabetes  
## 581 -0.3543379 -1.0518224 pos  
## 306 -0.8139199 -0.3872093 neg  
## 670 -0.9641678 0.9420168 neg  
## 283 -0.6754561 0.2774037 neg  
## 385 -0.7962436 -0.7195159 neg  
## 589 1.9524099 1.5235532 pos

head(test.norm.df)

## pregnant glucose pressure triceps insulin mass  
## 12 1.7414841 1.50548750 0.1703748 -0.05159703 -0.1780468 0.76633129  
## 17 -1.1349349 -0.13137610 1.0278408 1.95414666 1.1147467 1.87624801  
## 23 0.8785584 2.42213111 1.5423204 -0.05159703 -0.1780468 1.02246592  
## 24 1.4538422 -0.09863882 0.6848544 0.61698420 -0.1780468 -0.51434184  
## 30 0.3032746 -0.16411337 1.7138135 -0.05159703 -0.1780468 0.21137294  
## 38 1.4538422 -0.65517245 0.3418680 0.83984461 -0.1780468 0.04061652  
## pedigree age diabetes  
## 12 0.1347044 0.02817381 pos  
## 17 0.1759489 -0.22105608 pos  
## 23 -0.1186549 0.60971024 pos  
## 24 -0.6725100 -0.38720935 pos  
## 30 -0.4545032 0.36048034 neg  
## 38 0.5117973 1.02509340 pos

train.norm.df.x <- train.norm.df[, 1:8]; train.norm.df.y <- train.norm.df[, 9]  
valid.norm.df.x <- valid.norm.df[, 1:8]; valid.norm.df.y <- valid.norm.df[, 9]  
test.norm.df.x <- test.norm.df[, 1:8]; test.norm.df.y <- test.norm.df[, 9]

## Q3

library(FNN)  
knn.pred <- knn(train.norm.df.x, valid.norm.df.x,  
 cl = train.norm.df.y, k = 3)  
  
accuracy <- confusionMatrix(knn.pred,  
 as.factor(valid.norm.df.y))$overall[1]  
accuracy

## Accuracy   
## 0.7581699

## Q4

library(caret)  
  
accuracy.df <- data.frame(k=seq(1,460,1),accuracy=rep(0,460))  
head(accuracy.df)

## k accuracy  
## 1 1 0  
## 2 2 0  
## 3 3 0  
## 4 4 0  
## 5 5 0  
## 6 6 0

tail(accuracy.df)

## k accuracy  
## 455 455 0  
## 456 456 0  
## 457 457 0  
## 458 458 0  
## 459 459 0  
## 460 460 0

for(i in 1:460) {  
 knn.pred <- knn(train = train.norm.df.x,   
 test = valid.norm.df.x,  
 cl = train.norm.df.y,  
 k = i)  
 accuracy.df[i,2] <- confusionMatrix(knn.pred, as.factor(valid.norm.df.y))$overall[1]  
}

accuracy.df

## k accuracy  
## 1 1 0.6862745  
## 2 2 0.7254902  
## 3 3 0.7581699  
## 4 4 0.7712418  
## 5 5 0.7843137  
## 6 6 0.7647059  
## 7 7 0.8039216  
## 8 8 0.7908497  
## 9 9 0.8169935  
## 10 10 0.8169935  
## 11 11 0.8169935  
## 12 12 0.8169935  
## 13 13 0.8235294  
## 14 14 0.8104575  
## 15 15 0.8104575  
## 16 16 0.8300654  
## 17 17 0.7973856  
## 18 18 0.8169935  
## 19 19 0.8104575  
## 20 20 0.8039216  
## 21 21 0.8104575  
## 22 22 0.7973856  
## 23 23 0.7843137  
## 24 24 0.7843137  
## 25 25 0.7908497  
## 26 26 0.8039216  
## 27 27 0.7973856  
## 28 28 0.8104575  
## 29 29 0.8104575  
## 30 30 0.8169935  
## 31 31 0.8169935  
## 32 32 0.8169935  
## 33 33 0.7973856  
## 34 34 0.8039216  
## 35 35 0.8039216  
## 36 36 0.8104575  
## 37 37 0.8039216  
## 38 38 0.8039216  
## 39 39 0.8039216  
## 40 40 0.8039216  
## 41 41 0.8104575  
## 42 42 0.8039216  
## 43 43 0.7843137  
## 44 44 0.7908497  
## 45 45 0.7777778  
## 46 46 0.7843137  
## 47 47 0.7843137  
## 48 48 0.7973856  
## 49 49 0.7908497  
## 50 50 0.8039216  
## 51 51 0.7908497  
## 52 52 0.7908497  
## 53 53 0.7973856  
## 54 54 0.7973856  
## 55 55 0.7973856  
## 56 56 0.8039216  
## 57 57 0.7973856  
## 58 58 0.7908497  
## 59 59 0.7908497  
## 60 60 0.7973856  
## 61 61 0.7908497  
## 62 62 0.7908497  
## 63 63 0.7973856  
## 64 64 0.7973856  
## 65 65 0.7973856  
## 66 66 0.8039216  
## 67 67 0.7973856  
## 68 68 0.7973856  
## 69 69 0.7973856  
## 70 70 0.7973856  
## 71 71 0.8039216  
## 72 72 0.8235294  
## 73 73 0.8104575  
## 74 74 0.8235294  
## 75 75 0.8104575  
## 76 76 0.8235294  
## 77 77 0.8104575  
## 78 78 0.8104575  
## 79 79 0.8104575  
## 80 80 0.8235294  
## 81 81 0.8300654  
## 82 82 0.8300654  
## 83 83 0.8235294  
## 84 84 0.8235294  
## 85 85 0.8235294  
## 86 86 0.8235294  
## 87 87 0.8300654  
## 88 88 0.8169935  
## 89 89 0.8169935  
## 90 90 0.8169935  
## 91 91 0.8235294  
## 92 92 0.8235294  
## 93 93 0.8300654  
## 94 94 0.8300654  
## 95 95 0.8366013  
## 96 96 0.8300654  
## 97 97 0.8235294  
## 98 98 0.8300654  
## 99 99 0.8235294  
## 100 100 0.8300654  
## 101 101 0.8300654  
## 102 102 0.8300654  
## 103 103 0.8366013  
## 104 104 0.8300654  
## 105 105 0.8300654  
## 106 106 0.8300654  
## 107 107 0.8300654  
## 108 108 0.8300654  
## 109 109 0.8235294  
## 110 110 0.8235294  
## 111 111 0.8235294  
## 112 112 0.8235294  
## 113 113 0.8235294  
## 114 114 0.8235294  
## 115 115 0.8235294  
## 116 116 0.8235294  
## 117 117 0.8235294  
## 118 118 0.8169935  
## 119 119 0.8169935  
## 120 120 0.8169935  
## 121 121 0.8169935  
## 122 122 0.8169935  
## 123 123 0.8169935  
## 124 124 0.8169935  
## 125 125 0.8169935  
## 126 126 0.8169935  
## 127 127 0.8169935  
## 128 128 0.8235294  
## 129 129 0.8169935  
## 130 130 0.8104575  
## 131 131 0.8169935  
## 132 132 0.8169935  
## 133 133 0.8169935  
## 134 134 0.8104575  
## 135 135 0.8104575  
## 136 136 0.8169935  
## 137 137 0.8104575  
## 138 138 0.8104575  
## 139 139 0.8104575  
## 140 140 0.8235294  
## 141 141 0.8104575  
## 142 142 0.8235294  
## 143 143 0.8169935  
## 144 144 0.8235294  
## 145 145 0.8169935  
## 146 146 0.8169935  
## 147 147 0.8169935  
## 148 148 0.8169935  
## 149 149 0.8169935  
## 150 150 0.8169935  
## 151 151 0.8235294  
## 152 152 0.8169935  
## 153 153 0.8169935  
## 154 154 0.8169935  
## 155 155 0.8235294  
## 156 156 0.8235294  
## 157 157 0.8169935  
## 158 158 0.8104575  
## 159 159 0.8104575  
## 160 160 0.8104575  
## 161 161 0.8104575  
## 162 162 0.8104575  
## 163 163 0.8104575  
## 164 164 0.8039216  
## 165 165 0.8104575  
## 166 166 0.8039216  
## 167 167 0.8104575  
## 168 168 0.8169935  
## 169 169 0.8104575  
## 170 170 0.8104575  
## 171 171 0.8104575  
## 172 172 0.8104575  
## 173 173 0.8104575  
## 174 174 0.8169935  
## 175 175 0.8169935  
## 176 176 0.8104575  
## 177 177 0.8169935  
## 178 178 0.8104575  
## 179 179 0.8039216  
## 180 180 0.8039216  
## 181 181 0.8104575  
## 182 182 0.7973856  
## 183 183 0.7973856  
## 184 184 0.7973856  
## 185 185 0.7973856  
## 186 186 0.7973856  
## 187 187 0.7973856  
## 188 188 0.7908497  
## 189 189 0.7908497  
## 190 190 0.7908497  
## 191 191 0.7908497  
## 192 192 0.7908497  
## 193 193 0.7843137  
## 194 194 0.7908497  
## 195 195 0.7908497  
## 196 196 0.7908497  
## 197 197 0.7908497  
## 198 198 0.7908497  
## 199 199 0.7908497  
## 200 200 0.7843137  
## 201 201 0.7908497  
## 202 202 0.7843137  
## 203 203 0.7908497  
## 204 204 0.7843137  
## 205 205 0.7843137  
## 206 206 0.7777778  
## 207 207 0.7777778  
## 208 208 0.7712418  
## 209 209 0.7712418  
## 210 210 0.7777778  
## 211 211 0.7777778  
## 212 212 0.7647059  
## 213 213 0.7647059  
## 214 214 0.7647059  
## 215 215 0.7647059  
## 216 216 0.7712418  
## 217 217 0.7647059  
## 218 218 0.7712418  
## 219 219 0.7777778  
## 220 220 0.7777778  
## 221 221 0.7712418  
## 222 222 0.7712418  
## 223 223 0.7712418  
## 224 224 0.7777778  
## 225 225 0.7712418  
## 226 226 0.7777778  
## 227 227 0.7777778  
## 228 228 0.7712418  
## 229 229 0.7777778  
## 230 230 0.7647059  
## 231 231 0.7712418  
## 232 232 0.7777778  
## 233 233 0.7777778  
## 234 234 0.7712418  
## 235 235 0.7647059  
## 236 236 0.7647059  
## 237 237 0.7581699  
## 238 238 0.7581699  
## 239 239 0.7581699  
## 240 240 0.7581699  
## 241 241 0.7581699  
## 242 242 0.7516340  
## 243 243 0.7516340  
## 244 244 0.7581699  
## 245 245 0.7516340  
## 246 246 0.7581699  
## 247 247 0.7581699  
## 248 248 0.7581699  
## 249 249 0.7581699  
## 250 250 0.7516340  
## 251 251 0.7581699  
## 252 252 0.7581699  
## 253 253 0.7581699  
## 254 254 0.7581699  
## 255 255 0.7581699  
## 256 256 0.7516340  
## 257 257 0.7516340  
## 258 258 0.7516340  
## 259 259 0.7516340  
## 260 260 0.7516340  
## 261 261 0.7516340  
## 262 262 0.7516340  
## 263 263 0.7516340  
## 264 264 0.7516340  
## 265 265 0.7647059  
## 266 266 0.7450980  
## 267 267 0.7450980  
## 268 268 0.7450980  
## 269 269 0.7516340  
## 270 270 0.7516340  
## 271 271 0.7516340  
## 272 272 0.7385621  
## 273 273 0.7450980  
## 274 274 0.7385621  
## 275 275 0.7385621  
## 276 276 0.7320261  
## 277 277 0.7320261  
## 278 278 0.7254902  
## 279 279 0.7254902  
## 280 280 0.7254902  
## 281 281 0.7254902  
## 282 282 0.7254902  
## 283 283 0.7254902  
## 284 284 0.7189542  
## 285 285 0.7254902  
## 286 286 0.7189542  
## 287 287 0.7189542  
## 288 288 0.7189542  
## 289 289 0.7189542  
## 290 290 0.7124183  
## 291 291 0.7124183  
## 292 292 0.7124183  
## 293 293 0.7124183  
## 294 294 0.7124183  
## 295 295 0.7124183  
## 296 296 0.7124183  
## 297 297 0.7124183  
## 298 298 0.7124183  
## 299 299 0.7124183  
## 300 300 0.7124183  
## 301 301 0.7124183  
## 302 302 0.7124183  
## 303 303 0.7124183  
## 304 304 0.7124183  
## 305 305 0.7124183  
## 306 306 0.7058824  
## 307 307 0.7058824  
## 308 308 0.7058824  
## 309 309 0.7058824  
## 310 310 0.7058824  
## 311 311 0.7058824  
## 312 312 0.7058824  
## 313 313 0.7058824  
## 314 314 0.7058824  
## 315 315 0.7058824  
## 316 316 0.7058824  
## 317 317 0.7058824  
## 318 318 0.7058824  
## 319 319 0.7058824  
## 320 320 0.7058824  
## 321 321 0.7058824  
## 322 322 0.7058824  
## 323 323 0.7058824  
## 324 324 0.7058824  
## 325 325 0.7058824  
## 326 326 0.7058824  
## 327 327 0.7058824  
## 328 328 0.7058824  
## 329 329 0.7058824  
## 330 330 0.7058824  
## 331 331 0.7058824  
## 332 332 0.7058824  
## 333 333 0.7058824  
## 334 334 0.7058824  
## 335 335 0.7058824  
## 336 336 0.7058824  
## 337 337 0.7058824  
## 338 338 0.7058824  
## 339 339 0.7058824  
## 340 340 0.7058824  
## 341 341 0.7058824  
## 342 342 0.7058824  
## 343 343 0.7058824  
## 344 344 0.7058824  
## 345 345 0.7058824  
## 346 346 0.7058824  
## 347 347 0.7058824  
## 348 348 0.7058824  
## 349 349 0.7058824  
## 350 350 0.7058824  
## 351 351 0.7058824  
## 352 352 0.7058824  
## 353 353 0.7058824  
## 354 354 0.7058824  
## 355 355 0.7058824  
## 356 356 0.7058824  
## 357 357 0.7058824  
## 358 358 0.7058824  
## 359 359 0.7058824  
## 360 360 0.7058824  
## 361 361 0.7058824  
## 362 362 0.7058824  
## 363 363 0.7058824  
## 364 364 0.7058824  
## 365 365 0.7058824  
## 366 366 0.7058824  
## 367 367 0.7058824  
## 368 368 0.7058824  
## 369 369 0.7058824  
## 370 370 0.7058824  
## 371 371 0.7058824  
## 372 372 0.7058824  
## 373 373 0.7058824  
## 374 374 0.7058824  
## 375 375 0.7058824  
## 376 376 0.7058824  
## 377 377 0.7058824  
## 378 378 0.7058824  
## 379 379 0.7058824  
## 380 380 0.7058824  
## 381 381 0.7058824  
## 382 382 0.7058824  
## 383 383 0.7058824  
## 384 384 0.7058824  
## 385 385 0.7058824  
## 386 386 0.7058824  
## 387 387 0.7058824  
## 388 388 0.7058824  
## 389 389 0.7058824  
## 390 390 0.7058824  
## 391 391 0.7058824  
## 392 392 0.7058824  
## 393 393 0.7058824  
## 394 394 0.7058824  
## 395 395 0.7058824  
## 396 396 0.7058824  
## 397 397 0.7058824  
## 398 398 0.7058824  
## 399 399 0.7058824  
## 400 400 0.7058824  
## 401 401 0.7058824  
## 402 402 0.7058824  
## 403 403 0.7058824  
## 404 404 0.7058824  
## 405 405 0.7058824  
## 406 406 0.7058824  
## 407 407 0.7058824  
## 408 408 0.7058824  
## 409 409 0.7058824  
## 410 410 0.7058824  
## 411 411 0.7058824  
## 412 412 0.7058824  
## 413 413 0.7058824  
## 414 414 0.7058824  
## 415 415 0.7058824  
## 416 416 0.7058824  
## 417 417 0.7058824  
## 418 418 0.7058824  
## 419 419 0.7058824  
## 420 420 0.7058824  
## 421 421 0.7058824  
## 422 422 0.7058824  
## 423 423 0.7058824  
## 424 424 0.7058824  
## 425 425 0.7058824  
## 426 426 0.7058824  
## 427 427 0.7058824  
## 428 428 0.7058824  
## 429 429 0.7058824  
## 430 430 0.7058824  
## 431 431 0.7058824  
## 432 432 0.7058824  
## 433 433 0.7058824  
## 434 434 0.7058824  
## 435 435 0.7058824  
## 436 436 0.7058824  
## 437 437 0.7058824  
## 438 438 0.7058824  
## 439 439 0.7058824  
## 440 440 0.7058824  
## 441 441 0.7058824  
## 442 442 0.7058824  
## 443 443 0.7058824  
## 444 444 0.7058824  
## 445 445 0.7058824  
## 446 446 0.7058824  
## 447 447 0.7058824  
## 448 448 0.7058824  
## 449 449 0.7058824  
## 450 450 0.7058824  
## 451 451 0.7058824  
## 452 452 0.7058824  
## 453 453 0.7058824  
## 454 454 0.7058824  
## 455 455 0.7058824  
## 456 456 0.7058824  
## 457 457 0.7058824  
## 458 458 0.7058824  
## 459 459 0.7058824  
## 460 460 0.7058824

summary(accuracy.df$accuracy)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.6863 0.7059 0.7647 0.7606 0.8105 0.8366

accuracy.df[round(accuracy.df$accuracy,digits = 4)==0.8366,]

## k accuracy  
## 95 95 0.8366013  
## 103 103 0.8366013

accuracy.df[95,]

## k accuracy  
## 95 95 0.8366013

## Q5

knn.pred <- knn(train.norm.df.x,   
 test.norm.df.x,  
 cl = train.norm.df.y,   
 k = 95)  
result <- confusionMatrix(knn.pred,  
 as.factor(test.norm.df.y))$table  
result

## Reference  
## Prediction neg pos  
## neg 92 33  
## pos 9 21

accuracy <- confusionMatrix(knn.pred,as.factor(test.norm.df.y))$overall[1]  
accuracy

## Accuracy   
## 0.7290323