Performance Matrices

Ishaan Yadav

Import data

df<-read.csv("C:/Users/Admin/Downloads/Expt 6- Dataset\_wisconsinn breast cancerdata\_csv.csv")  
head(df)

## id diagnosis radius\_mean texture\_mean perimeter\_mean area\_mean  
## 1 842302 M 17.99 10.38 122.80 1001.0  
## 2 842517 M 20.57 17.77 132.90 1326.0  
## 3 84300903 M 19.69 21.25 130.00 1203.0  
## 4 84348301 M 11.42 20.38 77.58 386.1  
## 5 84358402 M 20.29 14.34 135.10 1297.0  
## 6 843786 M 12.45 15.70 82.57 477.1  
## smoothness\_mean compactness\_mean concavity\_mean concave.points\_mean  
## 1 0.11840 0.27760 0.3001 0.14710  
## 2 0.08474 0.07864 0.0869 0.07017  
## 3 0.10960 0.15990 0.1974 0.12790  
## 4 0.14250 0.28390 0.2414 0.10520  
## 5 0.10030 0.13280 0.1980 0.10430  
## 6 0.12780 0.17000 0.1578 0.08089  
## symmetry\_mean fractal\_dimension\_mean radius\_se texture\_se perimeter\_se  
## 1 0.2419 0.07871 1.0950 0.9053 8.589  
## 2 0.1812 0.05667 0.5435 0.7339 3.398  
## 3 0.2069 0.05999 0.7456 0.7869 4.585  
## 4 0.2597 0.09744 0.4956 1.1560 3.445  
## 5 0.1809 0.05883 0.7572 0.7813 5.438  
## 6 0.2087 0.07613 0.3345 0.8902 2.217  
## area\_se smoothness\_se compactness\_se concavity\_se concave.points\_se  
## 1 153.40 0.006399 0.04904 0.05373 0.01587  
## 2 74.08 0.005225 0.01308 0.01860 0.01340  
## 3 94.03 0.006150 0.04006 0.03832 0.02058  
## 4 27.23 0.009110 0.07458 0.05661 0.01867  
## 5 94.44 0.011490 0.02461 0.05688 0.01885  
## 6 27.19 0.007510 0.03345 0.03672 0.01137  
## symmetry\_se fractal\_dimension\_se radius\_worst texture\_worst perimeter\_worst  
## 1 0.03003 0.006193 25.38 17.33 184.60  
## 2 0.01389 0.003532 24.99 23.41 158.80  
## 3 0.02250 0.004571 23.57 25.53 152.50  
## 4 0.05963 0.009208 14.91 26.50 98.87  
## 5 0.01756 0.005115 22.54 16.67 152.20  
## 6 0.02165 0.005082 15.47 23.75 103.40  
## area\_worst smoothness\_worst compactness\_worst concavity\_worst  
## 1 2019.0 0.1622 0.6656 0.7119  
## 2 1956.0 0.1238 0.1866 0.2416  
## 3 1709.0 0.1444 0.4245 0.4504  
## 4 567.7 0.2098 0.8663 0.6869  
## 5 1575.0 0.1374 0.2050 0.4000  
## 6 741.6 0.1791 0.5249 0.5355  
## concave.points\_worst symmetry\_worst fractal\_dimension\_worst  
## 1 0.2654 0.4601 0.11890  
## 2 0.1860 0.2750 0.08902  
## 3 0.2430 0.3613 0.08758  
## 4 0.2575 0.6638 0.17300  
## 5 0.1625 0.2364 0.07678  
## 6 0.1741 0.3985 0.12440

scale the data

scaled\_df<-as.data.frame(scale(df[,3:32]))  
head(scaled\_df)

## radius\_mean texture\_mean perimeter\_mean area\_mean smoothness\_mean  
## 1 1.0960995 -2.0715123 1.2688173 0.9835095 1.5670875  
## 2 1.8282120 -0.3533215 1.6844726 1.9070303 -0.8262354  
## 3 1.5784992 0.4557859 1.5651260 1.5575132 0.9413821  
## 4 -0.7682333 0.2535091 -0.5921661 -0.7637917 3.2806668  
## 5 1.7487579 -1.1508038 1.7750113 1.8246238 0.2801253  
## 6 -0.4759559 -0.8346009 -0.3868077 -0.5052059 2.2354545  
## compactness\_mean concavity\_mean concave.points\_mean symmetry\_mean  
## 1 3.2806281 2.65054179 2.5302489 2.215565542  
## 2 -0.4866435 -0.02382489 0.5476623 0.001391139  
## 3 1.0519999 1.36227979 2.0354398 0.938858720  
## 4 3.3999174 1.91421287 1.4504311 2.864862154  
## 5 0.5388663 1.36980615 1.4272370 -0.009552062  
## 6 1.2432416 0.86554001 0.8239307 1.004517928  
## fractal\_dimension\_mean radius\_se texture\_se perimeter\_se area\_se  
## 1 2.2537638 2.4875451 -0.5647681 2.8305403 2.4853907  
## 2 -0.8678888 0.4988157 -0.8754733 0.2630955 0.7417493  
## 3 -0.3976580 1.2275958 -0.7793976 0.8501802 1.1802975  
## 4 4.9066020 0.3260865 -0.1103120 0.2863415 -0.2881246  
## 5 -0.5619555 1.2694258 -0.7895490 1.2720701 1.1893103  
## 6 1.8883435 -0.2548461 -0.5921406 -0.3210217 -0.2890039  
## smoothness\_se compactness\_se concavity\_se concave.points\_se symmetry\_se  
## 1 -0.2138135 1.31570389 0.7233897 0.66023900 1.1477468  
## 2 -0.6048187 -0.69231710 -0.4403926 0.25993335 -0.8047423  
## 3 -0.2967439 0.81425704 0.2128891 1.42357487 0.2368272  
## 4 0.6890953 2.74186785 0.8187979 1.11402678 4.7285198  
## 5 1.4817634 -0.04847723 0.8277425 1.14319885 -0.3607748  
## 6 0.1562093 0.44515196 0.1598845 -0.06906279 0.1340009  
## fractal\_dimension\_se radius\_worst texture\_worst perimeter\_worst area\_worst  
## 1 0.90628565 1.8850310 -1.35809849 2.3015755 1.9994782  
## 2 -0.09935632 1.8043398 -0.36887865 1.5337764 1.8888270  
## 3 0.29330133 1.5105411 -0.02395331 1.3462906 1.4550043  
## 4 2.04571087 -0.2812170 0.13386631 -0.2497196 -0.5495377  
## 5 0.49888916 1.2974336 -1.46548091 1.3373627 1.2196511  
## 6 0.48641784 -0.1653528 -0.31356043 -0.1149083 -0.2441054  
## smoothness\_worst compactness\_worst concavity\_worst concave.points\_worst  
## 1 1.3065367 2.6143647 2.1076718 2.2940576  
## 2 -0.3752817 -0.4300658 -0.1466200 1.0861286  
## 3 0.5269438 1.0819801 0.8542223 1.9532817  
## 4 3.3912907 3.8899747 1.9878392 2.1738732  
## 5 0.2203623 -0.3131190 0.6126397 0.7286181  
## 6 2.0467119 1.7201029 1.2621327 0.9050914  
## symmetry\_worst fractal\_dimension\_worst  
## 1 2.7482041 1.9353117  
## 2 -0.2436753 0.2809428  
## 3 1.1512420 0.2012142  
## 4 6.0407261 4.9306719  
## 5 -0.8675896 -0.3967505  
## 6 1.7525273 2.2398308

include the diagnosis columns in scaled\_df

scaled\_df$diagnosis<-df$diagnosis  
head(scaled\_df)

## radius\_mean texture\_mean perimeter\_mean area\_mean smoothness\_mean  
## 1 1.0960995 -2.0715123 1.2688173 0.9835095 1.5670875  
## 2 1.8282120 -0.3533215 1.6844726 1.9070303 -0.8262354  
## 3 1.5784992 0.4557859 1.5651260 1.5575132 0.9413821  
## 4 -0.7682333 0.2535091 -0.5921661 -0.7637917 3.2806668  
## 5 1.7487579 -1.1508038 1.7750113 1.8246238 0.2801253  
## 6 -0.4759559 -0.8346009 -0.3868077 -0.5052059 2.2354545  
## compactness\_mean concavity\_mean concave.points\_mean symmetry\_mean  
## 1 3.2806281 2.65054179 2.5302489 2.215565542  
## 2 -0.4866435 -0.02382489 0.5476623 0.001391139  
## 3 1.0519999 1.36227979 2.0354398 0.938858720  
## 4 3.3999174 1.91421287 1.4504311 2.864862154  
## 5 0.5388663 1.36980615 1.4272370 -0.009552062  
## 6 1.2432416 0.86554001 0.8239307 1.004517928  
## fractal\_dimension\_mean radius\_se texture\_se perimeter\_se area\_se  
## 1 2.2537638 2.4875451 -0.5647681 2.8305403 2.4853907  
## 2 -0.8678888 0.4988157 -0.8754733 0.2630955 0.7417493  
## 3 -0.3976580 1.2275958 -0.7793976 0.8501802 1.1802975  
## 4 4.9066020 0.3260865 -0.1103120 0.2863415 -0.2881246  
## 5 -0.5619555 1.2694258 -0.7895490 1.2720701 1.1893103  
## 6 1.8883435 -0.2548461 -0.5921406 -0.3210217 -0.2890039  
## smoothness\_se compactness\_se concavity\_se concave.points\_se symmetry\_se  
## 1 -0.2138135 1.31570389 0.7233897 0.66023900 1.1477468  
## 2 -0.6048187 -0.69231710 -0.4403926 0.25993335 -0.8047423  
## 3 -0.2967439 0.81425704 0.2128891 1.42357487 0.2368272  
## 4 0.6890953 2.74186785 0.8187979 1.11402678 4.7285198  
## 5 1.4817634 -0.04847723 0.8277425 1.14319885 -0.3607748  
## 6 0.1562093 0.44515196 0.1598845 -0.06906279 0.1340009  
## fractal\_dimension\_se radius\_worst texture\_worst perimeter\_worst area\_worst  
## 1 0.90628565 1.8850310 -1.35809849 2.3015755 1.9994782  
## 2 -0.09935632 1.8043398 -0.36887865 1.5337764 1.8888270  
## 3 0.29330133 1.5105411 -0.02395331 1.3462906 1.4550043  
## 4 2.04571087 -0.2812170 0.13386631 -0.2497196 -0.5495377  
## 5 0.49888916 1.2974336 -1.46548091 1.3373627 1.2196511  
## 6 0.48641784 -0.1653528 -0.31356043 -0.1149083 -0.2441054  
## smoothness\_worst compactness\_worst concavity\_worst concave.points\_worst  
## 1 1.3065367 2.6143647 2.1076718 2.2940576  
## 2 -0.3752817 -0.4300658 -0.1466200 1.0861286  
## 3 0.5269438 1.0819801 0.8542223 1.9532817  
## 4 3.3912907 3.8899747 1.9878392 2.1738732  
## 5 0.2203623 -0.3131190 0.6126397 0.7286181  
## 6 2.0467119 1.7201029 1.2621327 0.9050914  
## symmetry\_worst fractal\_dimension\_worst diagnosis  
## 1 2.7482041 1.9353117 M  
## 2 -0.2436753 0.2809428 M  
## 3 1.1512420 0.2012142 M  
## 4 6.0407261 4.9306719 M  
## 5 -0.8675896 -0.3967505 M  
## 6 1.7525273 2.2398308 M

split the data into training and testing sets

library(caTools)

## Warning: package 'caTools' was built under R version 4.3.3

set.seed(100)  
split\_vector<-sample.split(scaled\_df$diagnosis,SplitRatio=0.75)  
split\_vector

## [1] TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE  
## [13] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [25] TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE FALSE TRUE  
## [37] FALSE TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE FALSE FALSE FALSE  
## [49] TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE FALSE FALSE TRUE  
## [61] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE  
## [73] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [85] FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE FALSE TRUE  
## [97] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE  
## [109] TRUE TRUE FALSE FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE  
## [121] FALSE TRUE TRUE TRUE FALSE FALSE FALSE TRUE FALSE TRUE TRUE FALSE  
## [133] FALSE TRUE FALSE TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE  
## [145] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [157] FALSE TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [169] TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE  
## [181] TRUE TRUE FALSE FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE FALSE  
## [193] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE  
## [205] FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE FALSE FALSE TRUE TRUE  
## [217] TRUE TRUE FALSE TRUE TRUE FALSE TRUE FALSE FALSE TRUE TRUE TRUE  
## [229] FALSE TRUE TRUE TRUE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE  
## [241] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE  
## [253] TRUE FALSE FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE  
## [265] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [277] FALSE TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE  
## [289] TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [301] FALSE TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE  
## [313] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE  
## [325] TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE FALSE TRUE  
## [337] TRUE FALSE TRUE FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE  
## [349] TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE FALSE FALSE TRUE TRUE  
## [361] TRUE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [373] TRUE TRUE TRUE FALSE FALSE TRUE FALSE TRUE FALSE TRUE TRUE TRUE  
## [385] FALSE FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [397] FALSE FALSE FALSE TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE FALSE  
## [409] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE  
## [421] TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE FALSE TRUE TRUE FALSE  
## [433] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [445] FALSE FALSE TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [457] FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE FALSE TRUE  
## [469] TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE FALSE  
## [481] TRUE TRUE TRUE TRUE FALSE TRUE FALSE FALSE TRUE TRUE TRUE TRUE  
## [493] TRUE TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE  
## [505] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE  
## [517] TRUE FALSE FALSE TRUE TRUE FALSE FALSE FALSE FALSE TRUE TRUE TRUE  
## [529] TRUE FALSE FALSE TRUE TRUE FALSE TRUE TRUE FALSE FALSE FALSE FALSE  
## [541] TRUE TRUE TRUE FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE TRUE  
## [553] TRUE TRUE FALSE TRUE TRUE FALSE FALSE TRUE FALSE TRUE FALSE TRUE  
## [565] TRUE TRUE TRUE TRUE TRUE

train\_df<-subset(scaled\_df,split\_vector==T)  
test\_df<-subset(scaled\_df,split\_vector==F)

Apply KNN algorithm

library(class)

## Warning: package 'class' was built under R version 4.3.3

knn\_model<-knn(train\_df[,-31],test\_df[,-31],cl=train\_df$diagnosis,k=3)  
knn\_model

## [1] M M M M M M M M M B M M M B M B B B M B B B B B B B M B M M M B B M M B B  
## [38] M B M B B B M M B M M B M B M B B B M B B B M M M B B M B B M B B B B M B  
## [75] M M M B M B B B B B B B B B M B B B B M B B M M B B B B B M B B B B B B B  
## [112] M B B M B M B M B M B M B B B B B M M M B B B B B B B B B B M  
## Levels: B M

Save the prediction in the predictions column in test set

test\_df$predictions<-knn\_model  
head(test\_df)

## radius\_mean texture\_mean perimeter\_mean area\_mean smoothness\_mean  
## 7 1.16987830 0.1605082 1.1371245 1.09433201 -0.12302797  
## 12 0.46897996 -0.3254213 0.4786607 0.35835701 0.05259614  
## 30 0.97691843 -0.9857273 0.9478161 0.85307997 0.15000708  
## 31 1.27770882 1.3532442 1.3511252 1.23072892 0.71385291  
## 35 0.56829753 -0.3277463 0.6185842 0.43280699 0.54320600  
## 37 0.03482025 0.5650619 0.0683554 -0.06220013 0.13294239  
## compactness\_mean concavity\_mean concave.points\_mean symmetry\_mean  
## 7 0.08821762 0.2998086 0.6463664 -0.06426807  
## 12 0.47070096 0.1347304 0.4417422 0.11082315  
## 30 0.21508091 0.1248207 0.7888817 -0.26489343  
## 31 1.59732268 1.7950455 1.9452402 1.35470037  
## 35 0.97626062 0.5845559 0.7370814 0.67986962  
## 37 0.10336548 0.5406521 0.1819674 0.26767570  
## fractal\_dimension\_mean radius\_se texture\_se perimeter\_se area\_se  
## 7 -0.7616620 0.1497513 -0.8042322 0.1552737 0.2983649  
## 12 -0.2801003 0.3628681 -0.4204733 0.3451983 0.3038605  
## 30 -0.1852043 0.7036386 -0.7148637 0.8848019 0.4564182  
## 31 -0.1172192 1.5344695 0.4516402 1.3393351 1.4214441  
## 35 0.3331826 -0.2566491 -0.9628478 -0.3378380 -0.1166622  
## 37 -0.2191969 -0.4297389 -0.3586586 -0.1033997 -0.3391237  
## smoothness\_se compactness\_se concavity\_se concave.points\_se symmetry\_se  
## 7 -0.9082307 -0.6509952 -0.30986874 -0.22788851 -0.82893670  
## 12 -0.4229713 0.8449693 -0.13197205 0.16593446 -0.05592523  
## 30 -0.4709310 0.2709299 0.07209565 0.28262274 -0.15633205  
## 31 -0.2641046 0.4613457 0.66475331 -0.03502870 -0.05713496  
## 35 -0.9511946 0.1787932 -0.17404445 -0.44230323 -0.42489002  
## 37 -0.3873345 0.2497105 0.53853612 -0.03016669 -0.03173082  
## fractal\_dimension\_se radius\_worst texture\_worst perimeter\_worst area\_worst  
## 7 -0.610680472 1.36777980 0.3225990 1.3671224 1.2740985  
## 12 0.131929997 0.85880462 0.2607728 0.8701362 0.7348937  
## 30 -0.019993366 0.77397542 -1.0017841 0.8225207 0.6084353  
## 31 0.289144223 1.42364293 1.3557481 1.5843678 1.3865060  
## 35 0.008350545 0.81535552 0.2575188 0.7570495 0.6681518  
## 37 0.085823901 -0.07845455 0.7618907 0.2660152 -0.1422361  
## smoothness\_worst compactness\_worst concavity\_worst concave.points\_worst  
## 7 0.5181843 0.02119633 0.5091043 1.1956637  
## 12 0.3167164 1.94891190 0.5958631 1.0100626  
## 30 -0.3008262 0.17119332 -0.1116288 0.4715149  
## 31 0.7327913 1.08960710 1.6350519 1.0678728  
## 35 0.5357032 2.07285010 1.2233069 1.0922139  
## 37 0.5357032 1.07753108 1.1811258 0.4578230  
## symmetry\_worst fractal\_dimension\_worst diagnosis predictions  
## 7 0.2622449 -0.01471753 M M  
## 12 1.4405702 1.15463563 M M  
## 30 -0.2339771 -0.26331581 M M  
## 31 0.8780775 0.76817327 M M  
## 35 2.2131895 2.17892698 M M  
## 37 1.1156821 0.96638749 M M

Create a confusion matrix

confusion\_matrix<-table(Actual=test\_df$diagnosis,Predictions=test\_df$predictions)  
confusion\_matrix

## Predictions  
## Actual B M  
## B 87 2  
## M 1 52

Let the positive class be “Maligent” and negative class be “Benign” Evaluate accuracy

accuracy<-sum(diag(confusion\_matrix))/sum(confusion\_matrix)  
cat("Accuray of KNN model is",accuracy\*100,"%")

## Accuray of KNN model is 97.88732 %

Evaluate the sensitivity

sensitivity<-confusion\_matrix[2,2]/sum(confusion\_matrix[2,])  
cat("sensitivity of KNN model is",sensitivity\*100,"%")

## sensitivity of KNN model is 98.11321 %

Evaluate the specificity

specificity<-confusion\_matrix[1,1]/sum(confusion\_matrix[1,])  
cat("specificity of KNN model is",specificity\*100,"%")

## specificity of KNN model is 97.75281 %

Evaluate the precision

precision<-confusion\_matrix[2,2]/sum(confusion\_matrix[,2])  
cat("precision of KNN model is",precision\*100,"%")

## precision of KNN model is 96.2963 %