Module6L1

Neha Parulekar

March 10, 2016

# Additional packages needed

install.packages("ggplot2");  
install.packages("class");

require(ggplot2)

## Loading required package: ggplot2

require(class)

## Loading required package: class

* Go to the [UC Irvine Machine Learning Repository](https://archive.ics.uci.edu/ml/) and find a dataset for supervised classification. Every student MUST use a different dataset so you MUST get approved for which you can going to use. This can be the same dataset you used for the unsupervised clustering as long as the data has some labeled data.

## **About the dataset**

The dataset used is taken from UCI Machine learning Repository. The name of the dataset is **Yeast Dataset**. This dataset predicts the cellular localization sites of proteins. There are 8 attributes and 1484 instances. There are8 predictive and 1 outcome variables. The outcome class is localization sites.

Attribute Information. 1.Sequence Name: Accession number for the SWISS-PROT database 2.mcg: McGeoch's method for signal sequence recognition. 3.gvh: von Heijne's method for signal sequence recognition. 4.alm: Score of the ALOM membrane spanning region prediction program. 5.mit: Score of discriminant analysis of the amino acid content of the N-terminal region (20 residues long) of mitochondrial and non-mitochondrial proteins. 6.erl: Presence of "HDEL" substring (thought to act as a signal for retention in the endoplasmic reticulum lumen). Binary attribute. 7.pox: Peroxisomal targeting signal in the C-terminus. 8.vac: Score of discriminant analysis of the amino acid content of vacuolar and extracellular proteins. 9.nuc: Score of discriminant analysis of nuclear localization signals of nuclear and non-nuclear proteins.

# Loading the dataset   
  
data\_url <-"https://archive.ics.uci.edu/ml/machine-learning-databases/yeast/yeast.data"  
yeastdata <- read.csv(url(data\_url), header = FALSE, sep = "")  
head(yeastdata)

## V1 V2 V3 V4 V5 V6 V7 V8 V9 V10  
## 1 ADT1\_YEAST 0.58 0.61 0.47 0.13 0.5 0.0 0.48 0.22 MIT  
## 2 ADT2\_YEAST 0.43 0.67 0.48 0.27 0.5 0.0 0.53 0.22 MIT  
## 3 ADT3\_YEAST 0.64 0.62 0.49 0.15 0.5 0.0 0.53 0.22 MIT  
## 4 AAR2\_YEAST 0.58 0.44 0.57 0.13 0.5 0.0 0.54 0.22 NUC  
## 5 AATM\_YEAST 0.42 0.44 0.48 0.54 0.5 0.0 0.48 0.22 MIT  
## 6 AATC\_YEAST 0.51 0.40 0.56 0.17 0.5 0.5 0.49 0.22 CYT

yeastdata$V1 <- NULL  
yeastdata$V7 <- NULL  
head(yeastdata)

## V2 V3 V4 V5 V6 V8 V9 V10  
## 1 0.58 0.61 0.47 0.13 0.5 0.48 0.22 MIT  
## 2 0.43 0.67 0.48 0.27 0.5 0.53 0.22 MIT  
## 3 0.64 0.62 0.49 0.15 0.5 0.53 0.22 MIT  
## 4 0.58 0.44 0.57 0.13 0.5 0.54 0.22 NUC  
## 5 0.42 0.44 0.48 0.54 0.5 0.48 0.22 MIT  
## 6 0.51 0.40 0.56 0.17 0.5 0.49 0.22 CYT

## Assignment Questions

* Classify your data using k-Nearest Neighbors. Answer the following questions:

# getting the summary of the dataset  
summary(yeastdata)

## V2 V3 V4 V5   
## Min. :0.1100 Min. :0.1300 Min. :0.21 Min. :0.0000   
## 1st Qu.:0.4100 1st Qu.:0.4200 1st Qu.:0.46 1st Qu.:0.1700   
## Median :0.4900 Median :0.4900 Median :0.51 Median :0.2200   
## Mean :0.5001 Mean :0.4999 Mean :0.50 Mean :0.2612   
## 3rd Qu.:0.5800 3rd Qu.:0.5700 3rd Qu.:0.55 3rd Qu.:0.3200   
## Max. :1.0000 Max. :1.0000 Max. :1.00 Max. :1.0000   
##   
## V6 V8 V9 V10   
## Min. :0.5000 Min. :0.0000 Min. :0.0000 CYT :463   
## 1st Qu.:0.5000 1st Qu.:0.4800 1st Qu.:0.2200 NUC :429   
## Median :0.5000 Median :0.5100 Median :0.2200 MIT :244   
## Mean :0.5047 Mean :0.4999 Mean :0.2762 ME3 :163   
## 3rd Qu.:0.5000 3rd Qu.:0.5300 3rd Qu.:0.3000 ME2 : 51   
## Max. :1.0000 Max. :0.7300 Max. :1.0000 ME1 : 44   
## (Other): 90

length(yeastdata)

## [1] 8

table(yeastdata$V10)

##   
## CYT ERL EXC ME1 ME2 ME3 MIT NUC POX VAC   
## 463 5 35 44 51 163 244 429 20 30

print(yeastdata$V10)

## [1] MIT MIT MIT NUC MIT CYT MIT NUC MIT CYT NUC NUC CYT MIT MIT CYT CYT  
## [18] NUC NUC NUC CYT CYT CYT CYT CYT MIT CYT CYT MIT ME1 NUC MIT CYT NUC  
## [35] ME1 EXC MIT CYT MIT MIT ME2 ME2 ME2 ME3 VAC NUC ME2 EXC VAC ME3 CYT  
## [52] ME3 NUC CYT CYT CYT CYT CYT CYT MIT MIT NUC NUC NUC CYT CYT CYT CYT  
## [69] CYT EXC CYT CYT MIT MIT MIT MIT MIT MIT MIT MIT MIT MIT MIT MIT MIT  
## [86] ME3 ME3 EXC NUC CYT CYT MIT CYT NUC CYT ME3 EXC MIT MIT CYT ME3 ME3  
## [103] CYT NUC CYT ME3 CYT CYT CYT CYT POX NUC NUC NUC NUC MIT MIT MIT MIT  
## [120] MIT MIT MIT MIT MIT ME3 MIT NUC MIT NUC CYT CYT CYT CYT NUC NUC CYT  
## [137] ME3 NUC CYT NUC CYT NUC NUC NUC CYT NUC CYT CYT NUC NUC NUC CYT NUC  
## [154] CYT NUC NUC CYT NUC NUC MIT CYT CYT CYT NUC NUC ME3 ME3 ME3 ME3 ME3  
## [171] NUC CYT CYT CYT MIT POX NUC NUC NUC CYT CYT CYT CYT CYT CYT CYT CYT  
## [188] CYT CYT CYT ME1 MIT CYT MIT MIT MIT MIT MIT MIT MIT MIT MIT MIT MIT  
## [205] MIT MIT MIT MIT MIT MIT MIT MIT MIT CYT CYT CYT EXC MIT ME1 ERL VAC  
## [222] ME3 NUC CYT CYT NUC NUC NUC ME2 POX NUC NUC ME3 EXC EXC CYT CYT CYT  
## [239] MIT MIT MIT MIT MIT CYT CYT CYT MIT MIT ME3 ME3 POX NUC NUC NUC VAC  
## [256] NUC NUC NUC NUC NUC NUC CYT CYT NUC NUC MIT NUC NUC NUC ME2 CYT NUC  
## [273] ME3 CYT NUC NUC ME2 CYT CYT NUC ME3 ME1 CYT CYT CYT ME3 ME3 ME2 ME3  
## [290] CYT ME2 CYT ME2 CYT ME3 ME3 ME3 CYT ME2 CYT ME3 ME2 NUC NUC ERL EXC  
## [307] CYT CYT CYT CYT CYT CYT CYT CYT ME3 ME3 ME3 NUC EXC POX POX CYT ME1  
## [324] NUC ME3 ME1 ME2 CYT CYT CYT ME3 CYT ME3 ME1 CYT NUC CYT CYT CYT NUC  
## [341] ME3 NUC CYT NUC ME3 NUC ME1 NUC CYT CYT CYT CYT CYT CYT CYT CYT NUC  
## [358] NUC NUC NUC NUC NUC CYT ME2 CYT CYT CYT ME3 CYT CYT CYT CYT CYT NUC  
## [375] CYT NUC ME3 CYT ME2 CYT CYT CYT CYT NUC CYT NUC NUC CYT CYT NUC MIT  
## [392] CYT NUC NUC NUC NUC CYT NUC NUC MIT CYT CYT MIT CYT CYT NUC NUC NUC  
## [409] NUC NUC ME3 NUC NUC CYT ME3 ME3 NUC NUC NUC NUC NUC NUC NUC ME3 ME1  
## [426] NUC CYT NUC MIT CYT CYT ME3 MIT MIT CYT NUC NUC NUC NUC MIT CYT CYT  
## [443] ME3 ME3 ME3 ME3 ME3 ME3 ME3 ME3 POX MIT MIT CYT MIT MIT CYT MIT MIT  
## [460] MIT NUC MIT MIT MIT CYT NUC NUC CYT ME3 ME3 ME1 MIT MIT ME3 ME3 NUC  
## [477] ME3 ME2 NUC MIT ME3 ME3 ME1 ME2 ME3 ME3 CYT NUC ERL NUC NUC ME1 ME1  
## [494] MIT MIT EXC EXC ME3 ME3 NUC CYT CYT NUC ME1 CYT ME2 ERL ME3 EXC CYT  
## [511] NUC ME2 ME2 ME2 ME2 MIT ME3 CYT VAC NUC MIT ME2 ME2 CYT CYT NUC MIT  
## [528] ME3 MIT NUC MIT ME3 CYT NUC CYT NUC NUC NUC ME2 NUC CYT ME3 CYT NUC  
## [545] ME3 NUC CYT MIT MIT MIT MIT MIT NUC NUC NUC NUC MIT CYT POX MIT ME3  
## [562] ME2 CYT MIT NUC MIT MIT NUC EXC ME3 NUC NUC CYT CYT CYT NUC NUC CYT  
## [579] EXC EXC EXC MIT MIT MIT NUC NUC NUC MIT MIT MIT NUC CYT POX ME2 ME2  
## [596] ME2 MIT NUC MIT MIT NUC MIT MIT MIT MIT MIT MIT MIT MIT MIT MIT MIT  
## [613] MIT MIT MIT MIT MIT MIT MIT MIT MIT MIT MIT MIT MIT MIT MIT MIT MIT  
## [630] MIT MIT MIT MIT MIT ME1 MIT MIT MIT NUC NUC NUC MIT MIT NUC NUC NUC  
## [647] MIT MIT MIT MIT MIT MIT MIT MIT CYT MIT NUC NUC CYT CYT CYT EXC NUC  
## [664] EXC ME1 ME3 CYT NUC NUC MIT MIT CYT MIT NUC CYT CYT ME2 NUC NUC MIT  
## [681] NUC NUC NUC NUC NUC NUC CYT CYT NUC CYT NUC NUC NUC NUC NUC NUC NUC  
## [698] CYT MIT NUC NUC NUC NUC NUC NUC NUC NUC NUC ME2 ME2 MIT MIT MIT MIT  
## [715] NUC ME3 NUC NUC MIT ME2 POX CYT CYT NUC CYT POX POX POX POX CYT ME3  
## [732] MIT MIT CYT CYT CYT CYT CYT CYT ERL NUC NUC ME3 NUC MIT ME1 VAC VAC  
## [749] VAC VAC VAC NUC MIT MIT MIT MIT MIT MIT MIT MIT MIT MIT CYT CYT CYT  
## [766] CYT NUC CYT CYT CYT CYT NUC EXC EXC CYT EXC NUC EXC ME3 NUC ME3 NUC  
## [783] ME3 NUC MIT NUC MIT EXC EXC EXC ME3 ME1 ME3 ME3 VAC CYT ME2 ME2 ME3  
## [800] ME3 NUC ME3 ME3 NUC NUC NUC NUC NUC NUC NUC NUC NUC POX ME3 MIT CYT  
## [817] NUC CYT VAC VAC VAC CYT CYT CYT CYT CYT CYT NUC NUC CYT CYT NUC NUC  
## [834] NUC NUC NUC NUC NUC NUC NUC NUC NUC NUC NUC NUC CYT CYT CYT CYT CYT  
## [851] MIT NUC ME2 CYT CYT ME3 CYT CYT CYT MIT MIT NUC ME3 CYT CYT CYT MIT  
## [868] MIT MIT MIT MIT MIT NUC NUC NUC NUC NUC NUC NUC NUC NUC NUC NUC NUC  
## [885] NUC NUC NUC NUC NUC NUC NUC NUC NUC CYT CYT NUC CYT CYT NUC MIT NUC  
## [902] NUC NUC NUC NUC ME3 NUC NUC NUC NUC NUC NUC NUC NUC NUC CYT CYT NUC  
## [919] MIT MIT MIT NUC CYT NUC NUC NUC NUC NUC CYT CYT CYT CYT CYT CYT CYT  
## [936] CYT CYT CYT NUC NUC NUC NUC NUC NUC NUC NUC NUC NUC NUC NUC NUC NUC  
## [953] NUC NUC NUC NUC NUC NUC NUC NUC NUC CYT CYT CYT CYT CYT CYT CYT CYT  
## [970] CYT CYT CYT CYT CYT CYT CYT CYT CYT CYT CYT CYT CYT CYT CYT CYT CYT  
## [987] CYT CYT CYT CYT CYT CYT CYT CYT CYT CYT CYT CYT CYT CYT CYT MIT NUC  
## [1004] NUC MIT CYT CYT CYT CYT CYT CYT CYT CYT CYT CYT CYT CYT CYT CYT CYT  
## [1021] CYT CYT CYT CYT CYT CYT CYT CYT NUC NUC NUC CYT CYT ME3 CYT CYT ME1  
## [1038] CYT CYT MIT CYT ME3 MIT NUC MIT MIT MIT MIT CYT CYT ME1 ME2 CYT CYT  
## [1055] CYT CYT CYT CYT ME2 CYT ME2 CYT CYT CYT ME3 CYT ME3 ME3 ME3 CYT ME2  
## [1072] ME2 CYT CYT CYT MIT ME1 ME3 ME3 NUC NUC CYT NUC NUC VAC ME1 MIT MIT  
## [1089] CYT ME2 CYT NUC NUC NUC NUC NUC NUC NUC NUC CYT NUC NUC ME3 NUC NUC  
## [1106] CYT ME3 ME2 ME3 NUC CYT ME3 NUC NUC NUC MIT MIT CYT CYT ME3 ME3 CYT  
## [1123] NUC ME3 CYT NUC NUC NUC NUC NUC ME3 CYT MIT NUC NUC NUC CYT CYT CYT  
## [1140] NUC NUC NUC EXC EXC NUC POX NUC NUC NUC NUC NUC NUC NUC NUC NUC NUC  
## [1157] NUC NUC NUC NUC NUC NUC NUC NUC NUC NUC ME3 CYT CYT CYT CYT CYT CYT  
## [1174] CYT CYT CYT CYT CYT NUC MIT CYT NUC NUC ME3 ME3 EXC ME3 ME3 EXC NUC  
## [1191] ME3 ME3 CYT ME3 CYT ME1 CYT NUC ME3 MIT MIT NUC ME3 NUC CYT VAC NUC  
## [1208] EXC EXC EXC NUC CYT CYT CYT ME3 CYT CYT CYT CYT MIT NUC NUC NUC NUC  
## [1225] NUC ME1 ME3 CYT ME3 NUC CYT CYT CYT CYT CYT CYT CYT CYT CYT CYT NUC  
## [1242] NUC CYT CYT CYT CYT NUC NUC NUC NUC NUC NUC NUC CYT CYT CYT CYT CYT  
## [1259] CYT CYT CYT CYT CYT CYT CYT ME1 CYT ME1 ME1 CYT CYT NUC NUC NUC NUC  
## [1276] NUC NUC VAC CYT CYT CYT CYT CYT CYT CYT CYT CYT ME3 ME3 NUC MIT CYT  
## [1293] CYT CYT CYT CYT CYT CYT NUC ME2 CYT CYT CYT CYT MIT NUC NUC CYT CYT  
## [1310] CYT CYT CYT ME3 CYT CYT CYT CYT CYT CYT NUC ME3 NUC NUC NUC CYT NUC  
## [1327] CYT CYT CYT ME3 NUC VAC MIT CYT VAC VAC VAC VAC VAC VAC VAC VAC VAC  
## [1344] VAC VAC CYT CYT CYT CYT CYT CYT VAC CYT CYT ME1 NUC ME1 MIT ME1 NUC  
## [1361] ME3 ME3 NUC ME1 ME2 ME3 NUC CYT CYT ME1 ME3 ME3 MIT MIT NUC ME3 NUC  
## [1378] ME3 CYT MIT ME3 ME3 NUC ME3 NUC ME3 ME1 NUC POX POX ME3 MIT ME3 ME1  
## [1395] ME3 ME2 CYT MIT ME3 ME1 ME3 ME3 ME3 ME2 MIT ME3 ME3 ME3 ME1 NUC EXC  
## [1412] ME3 NUC ME3 CYT CYT ME3 ME3 NUC ME3 EXC ME3 NUC ME1 ME3 NUC ME3 ME3  
## [1429] ME3 ME3 ME3 ME3 ME1 ME3 ME2 NUC ME1 ME1 ME3 ME2 ME3 NUC POX ME1 MIT  
## [1446] ME3 CYT NUC NUC MIT ME1 NUC NUC ME3 NUC NUC ME1 CYT MIT CYT MIT POX  
## [1463] MIT NUC NUC NUC ME3 CYT MIT MIT MIT MIT NUC CYT CYT CYT CYT CYT CYT  
## [1480] ME2 NUC ME2 NUC CYT  
## Levels: CYT ERL EXC ME1 ME2 ME3 MIT NUC POX VAC

length(yeastdata$V10)

## [1] 1484

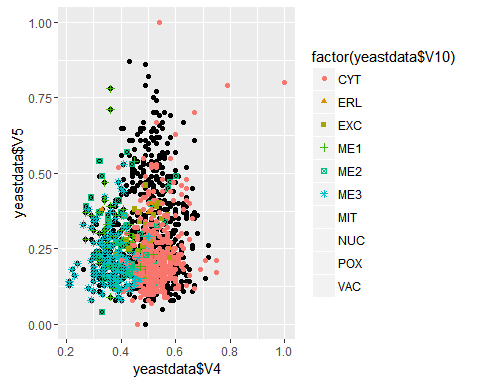
From the above we can see taht the columns range between 0.00 to 1.00 which is the min and max range. Thus the data does not require scaling or shuffling. Also the data seems to be properly mixed.

# plotting between V4,V5   
qplot(yeastdata$V4, yeastdata$V5, data = yeastdata) + geom\_point(aes(colour = factor(yeastdata$V10), shape = factor(yeastdata$V10)))

## Warning: The shape palette can deal with a maximum of 6 discrete values  
## because more than 6 becomes difficult to discriminate; you have  
## 10. Consider specifying shapes manually if you must have them.

## Warning: Removed 723 rows containing missing values (geom\_point).

## Warning: The shape palette can deal with a maximum of 6 discrete values  
## because more than 6 becomes difficult to discriminate; you have  
## 10. Consider specifying shapes manually if you must have them.

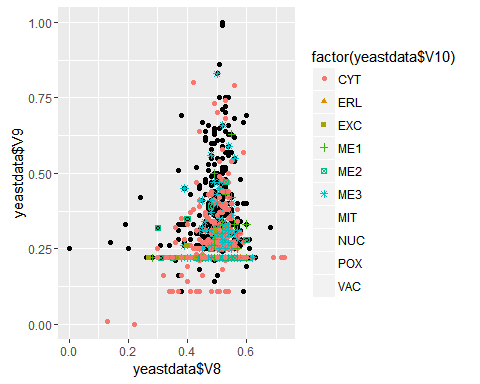


# plotting between V8,V9  
qplot(yeastdata$V8, yeastdata$V9, data = yeastdata) + geom\_point(aes(colour = factor(yeastdata$V10), shape = factor(yeastdata$V10)))

## Warning: The shape palette can deal with a maximum of 6 discrete values  
## because more than 6 becomes difficult to discriminate; you have  
## 10. Consider specifying shapes manually if you must have them.

## Warning: Removed 723 rows containing missing values (geom\_point).

## Warning: The shape palette can deal with a maximum of 6 discrete values  
## because more than 6 becomes difficult to discriminate; you have  
## 10. Consider specifying shapes manually if you must have them.



**Normalize the data and finding kNN value**

normalize <- function(x){  
 return((x-min(x)) / (max(x)-min(x)))  
}  
yeastdata.normalised <- as.data.frame(lapply(yeastdata[,c(1:7)], normalize))  
head(yeastdata.normalised)

## V2 V3 V4 V5 V6 V8 V9  
## 1 0.5280899 0.5517241 0.3291139 0.13 0 0.6575342 0.22  
## 2 0.3595506 0.6206897 0.3417722 0.27 0 0.7260274 0.22  
## 3 0.5955056 0.5632184 0.3544304 0.15 0 0.7260274 0.22  
## 4 0.5280899 0.3563218 0.4556962 0.13 0 0.7397260 0.22  
## 5 0.3483146 0.3563218 0.3417722 0.54 0 0.6575342 0.22  
## 6 0.4494382 0.3103448 0.4430380 0.17 0 0.6712329 0.22

summary(yeastdata.normalised)

## V2 V3 V4 V5   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.3371 1st Qu.:0.3333 1st Qu.:0.3165 1st Qu.:0.1700   
## Median :0.4270 Median :0.4138 Median :0.3797 Median :0.2200   
## Mean :0.4383 Mean :0.4252 Mean :0.3671 Mean :0.2612   
## 3rd Qu.:0.5281 3rd Qu.:0.5057 3rd Qu.:0.4304 3rd Qu.:0.3200   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## V6 V8 V9   
## Min. :0.000000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.000000 1st Qu.:0.6575 1st Qu.:0.2200   
## Median :0.000000 Median :0.6986 Median :0.2200   
## Mean :0.009434 Mean :0.6848 Mean :0.2762   
## 3rd Qu.:0.000000 3rd Qu.:0.7260 3rd Qu.:0.3000   
## Max. :1.000000 Max. :1.0000 Max. :1.0000

nrow(yeastdata.normalised)

## [1] 1484

# Training and testing   
yeastdata.normalised.train <- yeastdata.normalised[1:1250,]  
yeastdata.normalised.test <- yeastdata.normalised[1251:1484,]  
yeastdata.normalized.train.target <- yeastdata[1:1250,c(8)]  
yeastdata.normalized.test.target<- yeastdata[1251:1484,c(8)]

# Taking k value as 5  
k <- 5  
knn.m1 <- knn(train = yeastdata.normalised.train, test = yeastdata.normalised.test, yeastdata.normalized.train.target, k)  
length(knn.m1)

## [1] 234

# Taking k value as 7   
k <- 7  
knn.m2 <- knn(train = yeastdata.normalised.train, test = yeastdata.normalised.test, yeastdata.normalized.train.target, k)  
length(knn.m2)

## [1] 234

# Taking k value as 10  
k <- 10  
knn.m3 <- knn(train = yeastdata.normalised.train, test = yeastdata.normalised.test, yeastdata.normalized.train.target, k)  
length(knn.m3)

## [1] 234

# confusion matrix   
cm1 <- table(yeastdata.normalized.test.target, knn.m1)  
cm1

## knn.m1  
## yeastdata.normalized.test.target CYT ERL EXC ME1 ME2 ME3 MIT NUC POX VAC  
## CYT 48 0 0 0 0 0 6 20 2 0  
## ERL 0 0 0 0 0 0 0 0 0 0  
## EXC 1 0 0 1 0 0 0 0 0 0  
## ME1 0 0 1 16 2 0 0 0 0 0  
## ME2 1 0 1 1 2 0 2 1 0 0  
## ME3 6 0 0 0 1 28 3 7 0 0  
## MIT 5 0 0 0 0 0 7 7 0 0  
## NUC 13 0 0 0 0 0 2 30 1 1  
## POX 0 0 1 1 0 1 0 1 0 0  
## VAC 6 0 0 0 1 3 2 2 0 0

cm2 <- table(yeastdata.normalized.test.target, knn.m1)  
cm2

## knn.m1  
## yeastdata.normalized.test.target CYT ERL EXC ME1 ME2 ME3 MIT NUC POX VAC  
## CYT 48 0 0 0 0 0 6 20 2 0  
## ERL 0 0 0 0 0 0 0 0 0 0  
## EXC 1 0 0 1 0 0 0 0 0 0  
## ME1 0 0 1 16 2 0 0 0 0 0  
## ME2 1 0 1 1 2 0 2 1 0 0  
## ME3 6 0 0 0 1 28 3 7 0 0  
## MIT 5 0 0 0 0 0 7 7 0 0  
## NUC 13 0 0 0 0 0 2 30 1 1  
## POX 0 0 1 1 0 1 0 1 0 0  
## VAC 6 0 0 0 1 3 2 2 0 0

cm3 <- table(yeastdata.normalized.test.target, knn.m1)  
cm3

## knn.m1  
## yeastdata.normalized.test.target CYT ERL EXC ME1 ME2 ME3 MIT NUC POX VAC  
## CYT 48 0 0 0 0 0 6 20 2 0  
## ERL 0 0 0 0 0 0 0 0 0 0  
## EXC 1 0 0 1 0 0 0 0 0 0  
## ME1 0 0 1 16 2 0 0 0 0 0  
## ME2 1 0 1 1 2 0 2 1 0 0  
## ME3 6 0 0 0 1 28 3 7 0 0  
## MIT 5 0 0 0 0 0 7 7 0 0  
## NUC 13 0 0 0 0 0 2 30 1 1  
## POX 0 0 1 1 0 1 0 1 0 0  
## VAC 6 0 0 0 1 3 2 2 0 0

\***Does the k for kNN make a difference? Try for a range of values of k.** Using the k values of 5, 7 and 10, from the confusion matrix, we can see that on increasing the k value to 10, the predictions of CYT seems to be better than the other two. However there doesnt seem to be much difference in others, but this could be because of a large overlap of points in protien localization data in yeast.

\***Does scaling, normalization or leaving the data unscaled make a difference for kNN?** The yeast data looks properly mixed and also doesnt seem to be requiring scaling or shuffling as the columns range from 0.00 to 1.00. The data has to be normalized as without normalization there was infinity, not a number and or applicable error.