---

title: "math 425 homework 2 (rmarkdown) (2)"

author: "Mark Moreno"

date: "3/30/2020"

output: word\_document

---

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = TRUE)

```

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the \*\*Knit\*\* button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```{r cars}

summary(cars)

```

## Including Plots

You can also embed plots, for example:

```{r pressure, echo=FALSE}

plot(pressure)

```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

#import (wisc-train.csv)

#import (wisc-test.csv)

# KNN

library(caret)

wisc-train\_knn1=knn3(class~.,data = train,k=1)

pred\_knn1=predict(wisc-train\_knn1,test,type = 'class')

pred\_knn1 #predicted class

tab=table(pred\_knn5,test$class)

confusion\_knn1=prop.table(tab,2)

confusion\_knn1

err\_knn1=mean(pred\_knn1!=test$class) #classification error

err\_knn1

# LDA

library(MASS)

wisc-train\_lda=lda(class~.,data = train)

pred\_lda=predict(wisc-train\_lda,test)

pred\_lda$class #predicted classes

tab=table(pred\_lda$class,test$class)

confusion\_lda=prop.table(tab,2)

confusion\_lda

err\_lda=mean(pred\_lda$class!=test$class) #classification error

err\_lda

# QDA

library(MASS)

wisc-train\_qda=qda(class~.,data = train)

pred\_qda=predict(wisc-train\_qda,test)

pred\_qda$class #predicted class

tab=table(pred\_qda$class,test$class)

confusion\_qda=prop.table(tab,2)

confusion\_qda

err\_qda=mean(pred\_qda$class!=test$Species) #classification error

err\_qda

# train on multivariable logistic regression

wisc\_train<-glm(class~.,data=train)

summary(wisc\_train)

wisc\_prob<-plogis(predict(wisc\_train,data.frame(test)))

M\_pred<-rep("B",nrow(test)) ## all "B"

M\_pred[wisc-train\_prob>.5]<-"M" ## replace w/ "M" if above threshhold

tab=table(M\_pred,test$M)

prop.table(tab,2)

M\_pred<-rep("B",nrow(test))

M\_pred[wisc-train\_prob>.1]<-"M"

tab=table(M\_pred,test$M)

prop.table(tab,2)

M\_pred<-rep("B",nrow(test))

M\_pred[wisc-train\_prob>.9]<-"M"

tab=table(M\_pred,test$M)

prop.table(tab,2)

# Naive Bayes classifier

library(e1071)

wisc-train\_nb=naiveBayes(class~.,data = train)

pred\_nb=predict(wisc-train\_nb,test)

pred\_nb #gives class from predict()

tab=table(pred\_nb,test$class)

confusion\_nb=prop.table(tab,2)

confusion\_nb

err\_nb=mean(pred\_nb!=test$class) #classification error

err\_nb

####################################################################

#Repeat the above with only radius and symmetry as predictors

####################################################################

# KNN

library(caret)

wisc-train\_knn1=knn3(class~radius+symmetry,data = train,k=1)

pred\_knn1=predict(wisc-train\_knn1,test,type = 'class')

pred\_knn1 #predicted class

tab=table(pred\_knn5,test$class)

confusion\_knn1=prop.table(tab,2)

confusion\_knn1

err\_knn1=mean(pred\_knn1!=test$class) #classification error

err\_knn1

# LDA

library(MASS)

wisc-train\_lda=lda(class~radius+symmetry,data = train)

pred\_lda=predict(wisc-train\_lda,test)

pred\_lda$class #predicted classes

tab=table(pred\_lda$class,test$class)

confusion\_lda=prop.table(tab,2)

confusion\_lda

err\_lda=mean(pred\_lda$class!=test$class) #classification error

err\_lda

# QDA

library(MASS)

wisc-train\_qda=qda(class~radius+symmetry,data = train)

pred\_qda=predict(wisc-train\_qda,test)

pred\_qda$class #predicted class

tab=table(pred\_qda$class,test$class)

confusion\_qda=prop.table(tab,2)

confusion\_qda

err\_qda=mean(pred\_qda$class!=test$Species) #classification error

err\_qda

# train on multivariable logistic regression

wisc\_train<-glm(class~radius+symmetry,data=train)

summary(wisc\_train)

wisc\_prob<-plogis(predict(wisc\_train,data.frame(test)))

M\_pred<-rep("B",nrow(test)) ## all "B"

M\_pred[wisc-train\_prob>.5]<-"M" ## replace w/ "M" if above threshhold

tab=table(M\_pred,test$M)

prop.table(tab,2)

M\_pred<-rep("B",nrow(test))

M\_pred[wisc-train\_prob>.1]<-"M"

tab=table(M\_pred,test$M)

prop.table(tab,2)

M\_pred<-rep("B",nrow(test))

M\_pred[wisc-train\_prob>.9]<-"M"

tab=table(M\_pred,test$M)

prop.table(tab,2)

# Naive Bayes classifier

library(e1071)

wisc-train\_nb=naiveBayes(class~radius+symmetry,data = train)

pred\_nb=predict(wisc-train\_nb,test)

pred\_nb #gives class from predict()

tab=table(pred\_nb,test$class)

confusion\_nb=prop.table(tab,2)

confusion\_nb

err\_nb=mean(pred\_nb!=test$class) #classification error

err\_nb