~~~
---
title: "Todd_Garner_DS6306_Week7_FLS"
author: "Todd Garner"
date: "2023-02-12"
output: powerpoint_presentation
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
knitr::opts_chunk$set(dev = c('pdf', 'png'),
        fig.align = 'center', fig.height = 5, fig.width = 8.5,
        pdf.options(encoding = "ISOLatin9.enc"))

library(class)
library(caret)
library(e1071)
library(dplyr)
library(jsonlite)
library(ggplot2)
library(ggthemes)
library(tidyverse)
library(gridExtra)
~~~

DS 6306 Week 7 FLS

Part 1 - In the last unit you used a KNN classifier to classify the passengers who
survived and died. Now we will use a Naïve Bayes (NB) classifier and compare the two!

Question 1 - ***Using all 891 observations, train a NB model with Age and Pclass as
predictors and use this model to predict the survival of a 30 year old passenger in the 1,
2 and 3 classes. Use the "type = raw" option to look at the predicted percentage of each
outcome. (One slide.)***

```{r}
# Read in the training set.  Check to "View" the full file to make sure it's what we want.
Titanic <- read.csv(file.choose(), header = TRUE)
View(Titanic)
```
We won't need to run that piece of code again so I'm isolating it.

```{r}
Titanic$SurvivedF <- factor(Titanic$Survived, labels = c("Died", "Survived"))
head(Titanic$SurvivedF)

Titanic_sub <- Titanic %>% filter(!is.na(Age) & !is.na(Pclass))
dim(Titanic_sub)
Titanic_Sub_filter <- Titanic_sub %>% select(Age, Pclass, SurvivedF)
head(Titanic_Sub_filter)
dim(Titanic_Sub_filter)


model <- naiveBayes(Titanic_Sub_filter[((Titanic_Sub_filter$Age) &
(Titanic_Sub_filter$Pclass)), c("Age", "Pclass")], Titanic_Sub_filter$SurvivedF, laplace =
1)
df <- Titanic_Sub_filter %>% filter(Age == "30",Pclass == "1")
predict(model, df)
predict(model, df, type = "raw")
```
Question 2 - ***Split the 891 observations into a training and test set 70% - 30% using
this seed and code:***
      titanicClean = titanic %>% filter(!is.na(Age) & !is.na(Pclass))
      set.seed(4)
      *trainIndices =
```
```

~~~
sample(seq(1:length(titanicClean$Age)),round(.7length(titanicClean$Age)))*
      *trainTitanic = titanicClean[trainIndices,]*
      *testTitanic = titanicClean[-trainIndices,]*
      *(One slide that shows the head of trainTitanic and testTitanic)*


```{r}
Titanic_clean = Titanic %>% filter(!is.na(Age) & !is.na(Pclass))
set.seed(58)
trainIndices =
sample(seq(1:length(Titanic_clean$Age)),round(.7*length(Titanic_clean$Age)))
trainTitanic = Titanic_clean[trainIndices,]
testTitanic = Titanic_clean[-trainIndices,]
head(trainTitanic)
dim(trainTitanic)
head(testTitanic)
dim(testTitanic)
~~~
Question 3 - ***Train a NB model based on the training set using just the Age and Pclass
variables. Use the model to predict the survival of those in the test set and use those
results to evaluate the model based on accuracy, sensitivity and specificity. Finally,
Compare the results to what you found with the KNN classifier. (At least one slide.)***

```{r}
model <- naiveBayes(trainTitanic,as.factor(trainTitanic$Survived) , laplace = 1)
summary(model)
dim(model)
#head(model)
df <- data.frame(testTitanic)
nrow(df)
x <- round(predict(model, df, type = "raw"), digits = 1)
y <- x[,2]
y


dim(df$Survived)
nrow(df$Survived)
table(y, df$Survived)
confusionMatrix(table(y, df$Survived))

#table(count_("Died"), count("Survived"))
```

Question 4 - ***Now repeat the above with a new seed and compare the accuracy, sensitivity
and specificity. Do this 3 or 4 times to observe the variance in the statistics. (At
least one slide.)***

###By changing the seed, the metrics in the confusion matrix changed. Likely because
there were more or less NA's in each instance, but the accuracy never wavered much away
from 100%. I must say this is surprising as it just doesn't seem likely to have a model
that is 100% accurate. I checked and rechecked my model and my data.frame and made sure
that the model was fed by training data via the model and testing data via the other
variable in the table/confusionMatrix. 100% sure made me think I was comparing train to
train or test to test. I still have a nagging feeling that I've missed something
somewhere.

Question 5 - ***Write a loop to repeat the above for 100 different values of the seed.
Find the average of the accuracy, sensitivity and specificity to get a stable (smaller
variance) statistic to evaluate the model. (At least one slide.)***

```{r}
#a = c()
#b = c()
iterations = 100
Titanic_clean = Titanic %>% filter(!is.na(Age) & !is.na(Pclass))
```

~~~
for(i in 1:iterations) {
set.seed(i)
trainIndices =
sample(seq(1:length(Titanic_clean$Age)),round(.7*length(Titanic_clean$Age)))
trainTitanic = Titanic_clean[trainIndices,]
testTitanic = Titanic_clean[-trainIndices,]

model <- naiveBayes(trainTitanic,as.factor(trainTitanic$Survived) , laplace = 1)
df <- data.frame(testTitanic)
x <- round(predict(model, df, type = "raw"), digits = 1)
y <- x[,2]
#table(y, df$Survived)
#confusionMatrix(table(y, df$Survived))
#a <- data.frame(a)
#b <- data.frame(b)
#a <- append(a[i,]) <- sensitivity(factor(y), factor(df$Survived))
#b <- append(b[i,]) <- specificity(factor(y), factor(df$Survived))
master_sens[,i] = sensitivity(factor(y), factor(df$Survived))
master_spec[,i] = specificity(factor(y), factor(df$Survived))
}

mean_sens = colMeans(master_sens)
mean_spec = colMeans(master_spec)

which.max(mean_sens)
max(mean_sens)
which.max(mean_spec)
max(mean_spec)
```
--------------------------------
set.seed(1)
iterations = 10
numks = 20

masterAcc = matrix(nrow = iterations, ncol = numks)

for(j in 1:iterations)
{

  for(i in 1:numks)
  {
    CM = confusionMatrix(table(iris[,5],knn.cv(iris[,c(1,2)],iris[,5],k = i)))
    masterAcc[j,i] = CM$overall[1]

  }

}

MeanAcc = colMeans(masterAcc)

plot(seq(1,numks,1),MeanAcc, type = "l")

which.max(MeanAcc)
max(MeanAcc)
```
~~~