TODD GARNER

DS 6306 WEEK 7 PART 1

FEBRUARY 14. 2023

PART 1, #1

- Read in the data in the first chunk, so I don't have to continually reload the .csv:

```
32
33 ```{r}
34 # Read in the training set.  Check to "View" the full file to make sure it's what we want.
35 Titanic <- read.csv(file.choose(), header = TRUE)
36 View(Titanic)
37 ```
```

- Find the probability that a 30 y/o survived in each Pclass: 1, 2, 3

```
```{r}
Titanic$SurvivedF <- factor(Titanic$Survived, labels = c("Died", "Survived"))
Titanic_sub <- Titanic %>% filter(!is.na(Age) & !is.na(Pclass))
Titanic_Sub_filter <- Titanic_sub %>% select(Age, Pclass, SurvivedF)

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")
```
```

- Results:

Age	Pclass	Survived?	Died/Survived
30	1	Yes	.2898412, .7101588
30	2	No	.5877154, .4122846
30	3	No	.7730445, .2269555

P1 #2, SPLIT 891 INTO 70%/30% TRAIN/TEST SET

- head(trainTitanic)

	PassengerId <int>	Survived <int>	Pclass <int>	Name <chr>	Sex <chr>	Age <dbl>	SibSp <int>	Parch <int>	Ticket <chr>
504	637	0	3	Leinonen, Mr. Antti Gustaf	male	32	0	0	STON/O 2. 3101
587	737	0	3	Ford, Mrs. Edward (Margaret Ann Watson)	female	48	1	3	W./C. 6608
71	92	0	3	Andreasson, Mr. Paul Edvin	male	20	0	0	347466
684	856	1	3	Aks, Mrs. Sam (Leah Rosen)	female	18	0	1	392091
371	463	0	1	Gee, Mr. Arthur H	male	47	0	0	111320
698	873	0	1	Carlsson, Mr. Frans Olof	male	33	0	0	695

6 rows | 1-10 of 13 columns

	Sex <chr>	Age <dbl>	SibSp <int>	Parch <int>	Ticket <chr>	Fare <dbl>	Cabin <chr>	Embarked <chr>	SurvivedF <fctr>
504	male	32	0	0	STON/O 2. 3101292	7.9250		S	Died
587	female	48	1	3	W./C. 6608	34.3750		S	Died
71	male	20	0	0	347466	7.8542		S	Died
684	female	18	0	1	392091	9.3500		S	Survived
371	male	47	0	0	111320	38.5000	E63	S	Died
698	male	33	0	0	695	5.0000	B51 B53 B55	S	Died

rows | 6-14 of 13 columns

- head(testTitanic)

	PassengerId <int>	Survived <int>	Pclass <int>	Name <chr>	Sex <chr>	Age <dbl>	SibSp <int>	Parch <int>	Ticket <chr>
1	1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171
6	7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463
7	8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909
12	13	0	3	Saundercock, Mr. William Henry	male	20	0	0	A/5. 2151
16	17	0	3	Rice, Master. Eugene	male	2	4	1	382652
17	19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele)	female	31	1	0	345763

6 rows | 1-9 of 13 columns

	Fare <dbl>	Cabin <chr>	Embarked <chr>	SurvivedF <fctr>
1	7.2500		S	Died
6	51.8625	E46	S	Died
7	21.0750		S	Died
12	8.0500		S	Died
16	29.1250		Q	Died
17	18.0000		S	Died

rows | 6-14 of 13 columns

P1 #3, TRAIN NB MODEL AND COMPARE AGAINST KNN

Last week's results: KNN

```
Confusion Matrix and Statistics

   knn.24
     0  1
  0 69 10
  1 46 18

               Accuracy : 0.6084
                 95% CI : (0.5233, 0.6889)
    No Information Rate : 0.8042
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1634

 Mcnemar's Test P-Value : 2.91e-06

            Sensitivity : 0.6000
            Specificity : 0.6429
         Pos Pred Value : 0.8734
         Neg Pred Value : 0.2813
             Prevalence : 0.8042
         Detection Rate : 0.4825
   Detection Prevalence : 0.5524
      Balanced Accuracy : 0.6214

       'Positive' Class : 0
```

100% Accuracy certainly gives me concern. Any model that provides 100% Accuracy should. I checked and rechecked my inputs and I cannot find an error.

This week's results: Naïve Bayes

```
Confusion Matrix and Statistics


y     0   1
  0 128   0
  1   0  86

               Accuracy : 1
                 95% CI : (0.9829, 1)
    No Information Rate : 0.5981
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1

 Mcnemar's Test P-Value : NA

            Sensitivity : 1.0000
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 1.0000
             Prevalence : 0.5981
         Detection Rate : 0.5981
   Detection Prevalence : 0.5981
      Balanced Accuracy : 1.0000

       'Positive' Class : 0
```

P1 #4, VARYING THE SEED NUMEROUS TIMES ON NB MODEL

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.

Seed: 4

```
Confusion Matrix and Statistics


y     0   1
  0 128   0
  1   0  86

               Accuracy : 1
                 95% CI : (0.9829, 1)
    No Information Rate : 0.5981
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1

 Mcnemar's Test P-Value : NA

            Sensitivity : 1.0000
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 1.0000
             Prevalence : 0.5981
         Detection Rate : 0.5981
   Detection Prevalence : 0.5981
      Balanced Accuracy : 1.0000

       'Positive' Class : 0
```

Seed: 53

```
Confusion Matrix and Statistics


y     0   1
  0 117   0
  1   1  96

               Accuracy : 0.9953
                 95% CI : (0.9742, 0.9999)
    No Information Rate : 0.5514
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9906

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9915
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9897
             Prevalence : 0.5514
         Detection Rate : 0.5467
   Detection Prevalence : 0.5467
      Balanced Accuracy : 0.9958

       'Positive' Class : 0
```

Seed: 147

```
Confusion Matrix and Statistics


y     0   1
  0 131   0
  1   1  82

               Accuracy : 0.9953
                 95% CI : (0.9742, 0.9999)
    No Information Rate : 0.6168
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9901

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9924
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9880
             Prevalence : 0.6168
         Detection Rate : 0.6121
   Detection Prevalence : 0.6121
      Balanced Accuracy : 0.9962

       'Positive' Class : 0
```

Seed: 11

```
Confusion Matrix and Statistics


y     0   1
  0 136   0
  1   2  76

               Accuracy : 0.9907
                 95% CI : (0.9666, 0.9989)
    No Information Rate : 0.6449
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9797

 Mcnemar's Test P-Value : 0.4795

            Sensitivity : 0.9855
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9744
             Prevalence : 0.6449
         Detection Rate : 0.6355
   Detection Prevalence : 0.6355
      Balanced Accuracy : 0.9928

       'Positive' Class : 0
```

P1 #5, WRITE A LOOP FOR 100 DIFFERENT SEED VALUES – OBTAIN MEAN ACCURACY, SENSITIVITY, SPECIFICITY

- Code:

```r
```{r}
Titanic_clean = Titanic %>% filter(!is.na(Age) & !is.na(Pclass))
iterations = 100
master_sens <- 0
master_spec <- 0
master_acc <- 0
master_sens <- data.frame(master_sens)
master_spec <- data.frame(master_spec)
master_acc <- data.frame(master_acc)

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$Survived) , laplace = 1)
df <- data.frame(testTitanic)
x <- round(predict(model, df, type = "raw"), digits = 0)
y <- x[,2]
y
master_sens[,i] = sensitivity(factor(y), factor(df$Survived))
master_spec[,i] = specificity(factor(y), factor(df$Survived))
z <- table(factor(y), factor(df$Survived))
CM <- confusionMatrix(z, k = i)
master_acc[,i] = CM$overall[1]

}
mean_sens = colMeans(master_sens)
mean_spec = colMeans(master_spec)
mean_acc = colMeans(master_acc)

which.max(mean_sens)
max(mean_sens)
which.max(mean_spec)
max(mean_spec)
which.max(mean_acc)
max(mean_acc)


plot(mean_sens,xlab = "Iterations", ylab = "Mean Sensitivity", main = "Seed iterations versus Mean Sensitivity", type = "b")
plot(mean_spec,xlab = "Iterations", ylab = "Mean Specificity", main = "Seed iterations versus Mean Specificity",type = "b")
plot(mean_acc, xlab = "Iterations", ylab = "Mean Accuracy", main = "Seed iterations versus Mean Accuracy",type = "b")
```
```

P1 #5, RESULTS: MEAN VALUES, PLOTS OF EACH MEAN

- In each instance, accuracy, specificity, and sensitivity: Mean = 1