FLS 7 Haitie Liu

Titanic Study Naïve Bays 1.

```
Titanic=read.csv(file.choose(),header = TRUE)
TitanicNB=naiveBayes(Titanic[,c(3,6)],Titanic$Survived)
df1=data.frame(Pclass=1,Age=30)
df2=data.frame(Pclass=2,Age=30)
df3=data.frame(Pclass=3,Age=30)
Titanic$Age=as.numeric(Titanic$Age)
Titanic$Pclass=as.numeric(Titanic$Pclass)
predict(TitanicNB,df1,Titanic$Survived,type = "raw")
predict(TitanicNB,df2,Titanic$Survived,type = "raw")
predict(TitanicNB,df3,Titanic$Survived,type = "raw")
```

```
> predict(TitanicNB,df1,Titanic$Survived,type = "raw")
          0         1
[1,] 0.2953416 0.7046584
> predict(TitanicNB,df2,Titanic$Survived,type = "raw")
          0         1
[1,] 0.6063917 0.3936083
> predict(TitanicNB,df3,Titanic$Survived,type = "raw")
          0         1
[1,] 0.7735716 0.2264284
```

Presented to the left

After running naive bay on the data set,

For P class equal to 1,2,3 age equal to 30

All cases have tested "0" with probability greater than "50", which means the survivability is very low for all three classes

Titanic Study Naïve Bays 2.

```
Confusion Matrix and Statistics


result   0   1
     0 108  38
     1  20  48

               Accuracy : 0.729
                 95% CI : (0.6642, 0.7873)
    No Information Rate : 0.5981
    P-Value [Acc > NIR] : 4.364e-05

                  Kappa : 0.4162

 Mcnemar's Test P-Value : 0.0256

            Sensitivity : 0.8438
            Specificity : 0.5581
         Pos Pred Value : 0.7397
         Neg Pred Value : 0.7059
             Prevalence : 0.5981
         Detection Rate : 0.5047
   Detection Prevalence : 0.6822
      Balanced Accuracy : 0.7009

       'Positive' Class : 0
```

```
Confusion Matrix and Statistics


C1    0   1
  0 110  47
  1  18  39

               Accuracy : 0.6963
                 95% CI : (0.6299, 0.7571)
    No Information Rate : 0.5981
    P-Value [Acc > NIR] : 0.0018596

                  Kappa : 0.3312

 Mcnemar's Test P-Value : 0.0005147

            Sensitivity : 0.8594
            Specificity : 0.4535
         Pos Pred Value : 0.7006
         Neg Pred Value : 0.6842
             Prevalence : 0.5981
         Detection Rate : 0.5140
   Detection Prevalence : 0.7336
      Balanced Accuracy : 0.6564

       'Positive' Class : 0
```

Confusion Matrix using Naive Bays on the left.

Confusion Matrix using KNN, k=15 on the right

We can see that using Naive Bays has slightly higher accuracy and specificity, but lower in sensitivity

Titanic Study Naïve Bays 3.

```
for (i in 1:4){

  titanicClean = Titanic %>% filter(!is.na(Age) & !is.na(Pclass))

  trainIndices = sample(seq(1:length(titanicClean$Age)),round(.7*length(titanicClean$Age)))

  trainTitanic = titanicClean[trainIndices,]

  testTitanic = titanicClean[-trainIndices,]

model=naiveBayes(trainTitanic[,c(3,6)],trainTitanic$Survived)
result=predict(model,testTitanic[,c(3,6)],testTitanic$Survived,type = NULL)
CM=confusionMatrix(table(result,testTitanic$Survived))
CMN[i]=CM$overall[1]
CMSEN[i]=CM$byClass[1]
CMSPE[i]=CM$byClass[2]
}

sets=data.frame(CMN,CMSEN,CMSPE)

var(sets$CMN)
var(sets$CMSEN)
var(sets$CMSPE)
```

```
> var(sets$CMN)
[1] 0.001242831
> var(sets$CMSEN)
[1] 0.0007589995
> var(sets$CMSPE)
[1] 0.00344503
> sets
        CMN      CMSEN      CMSPE
1 0.6588785 0.8250000 0.4468085
2 0.6822430 0.8888889 0.4329897
3 0.6588785 0.8412698 0.3977273
4 0.7336449 0.8615385 0.5357143
```

Running the same Naïve Bays four time, using 70/30 split of the data sets

Accuracy as CMN,
Sensitivity as SEN,
Specificity as SPE,
We can see that each time there is slight variance between them, if we calculate the variance to each column

We get 0.00124 for Accuracy,
0.00076 for sensitivity
0.00344 for specificity

Titanic Study Naïve Bays 4.

```
CMN=numeric(100)
CMSEN=numeric(100)
CMSPE=numeric(100)

for (i in 1:100){

  titanicClean = Titanic %>% filter(!is.na(Age) & !is.na(Pclass))

  trainIndices = sample(seq(1:length(titanicClean$Age)),round(.7*length(titanicClean$Age)))

  trainTitanic = titanicClean[trainIndices,]

  testTitanic = titanicClean[-trainIndices,]

  model=naiveBayes(trainTitanic[,c(3,6)],trainTitanic$Survived)
  result=predict(model,testTitanic[,c(3,6)],testTitanic$Survived,type = NULL)
  CM=confusionMatrix(table(result,testTitanic$Survived))
  CMN[i]=CM$overall[1]
  CMSEN[i]=CM$byClass[1]
  CMSPE[i]=CM$byClass[2]
}

sets=data.frame(CMN,CMSEN,CMSPE)

var(sets$CMN)
var(sets$CMSEN)
var(sets$CMSPE)

mean(sets$CMN)
mean(sets$CMSEN)
mean(sets$CMSPE)
```

```
> var(sets$CMN)
[1] 0.0007800492
> var(sets$CMSEN)
[1] 0.0006902531
> var(sets$CMSPE)
[1] 0.002522698
>
> mean(sets$CMN)
[1] 0.697243
> mean(sets$CMSEN)
[1] 0.84932
> mean(sets$CMSPE)
[1] 0.4689064
```

Using for loops to run same experiment for 100 times and saving them in difference vector.

We get different averages for Accuracy, sensitivity, specificity as shown

Accuracy:0.6972

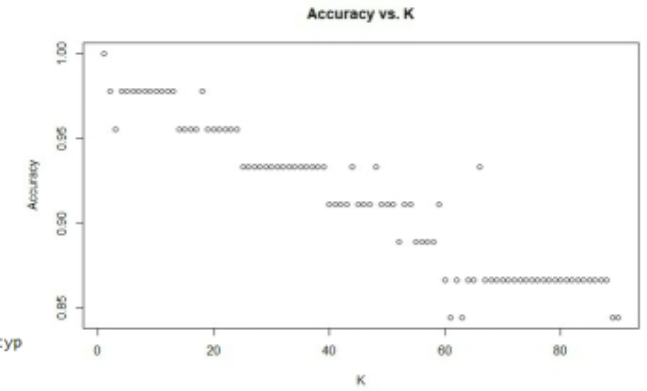
Sensitivity:0.8493

Specificity: 0.4689

With variance clearly lower than the previous slide shown

IRIS NAIVE BAYS STUDY

```
iris

indices=sample(1:dim(iris)[1],0.7*round(dim(iris)[1]))

iristrain=iris[indices,]
iristest=iris[-indices,]


CMA=numeric(100)
CMSEN=numeric(100)
CMSPE=numeric(100)

for (i in 1:100){
  indices=sample(1:dim(iris)[1],0.7*round(dim(iris)[1]))

  iristrain=iris[indices,]
  iristest=iris[-indices,]

  model=naiveBayes(iristrain[,c(1,2)],iristrain$Species)
  result=predict(model,iristest[,c(1,2)],iristest$Species,typ
  CM=confusionMatrix(table(result,iristest$Species))
  CMA[i]=CM$overall[1]
  CMSEN[i]=CM$byClass[1]
  CMSPE[i]=CM$byClass[2]
}

df=data.frame(CMA,CMSEN,CMSPE)
mean(df$CMA)
mean(df$CMSEN)
mean(df$CMSPE)
```




```
> mean(df$CMA)
[1] 0.7857778
> mean(df$CMSEN)
[1] 0.9775693
> mean(df$CMSPE)
[1] 0.7250935
```

Using Iris data doing 70/30 split looping 100 times, the average accuracy, sensitivity, and specificity is shown below.
Accuracy: 0.7856
Sensitivity:0.9776
Specificity:0.7250,

Comparing with unit 6, we can see that the accuracy is clearly higher using KNN

SEARCH TRUMP IN NYT CLASSIFIER

```
term <- "Trump"
begin_date <- "20230202"
end_date <- "20230212"
```

```
Confusion Matrix and Statistics

          Reference
Prediction News Other
     News      3     0
     Other     4     5

               Accuracy : 0.6667
                 95% CI : (0.3489, 0.9008)
    No Information Rate : 0.5833
    P-Value [Acc > NIR] : 0.3916

                  Kappa : 0.3846

 Mcnemar's Test P-Value : 0.1336

            Sensitivity : 0.4286
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.5556
             Prevalence : 0.5833
         Detection Rate : 0.2500
   Detection Prevalence : 0.2500
      Balanced Accuracy : 0.7143

       'Positive' Class : News
```

```
Confusion Matrix and Statistics

          Reference
Prediction News Other
     News     25     2
     Other     0    13

               Accuracy : 0.95
                 95% CI : (0.8308, 0.9939)
    No Information Rate : 0.625
    P-Value [Acc > NIR] : 2.092e-06

                  Kappa : 0.8904

 Mcnemar's Test P-Value : 0.4795

            Sensitivity : 1.0000
            Specificity : 0.8667
         Pos Pred Value : 0.9259
         Neg Pred Value : 1.0000
             Prevalence : 0.6250
         Detection Rate : 0.6250
   Detection Prevalence : 0.6750
      Balanced Accuracy : 0.9333

       'Positive' Class : News
```

Searching "Trump" between 2023/02/02-2023/02/12.

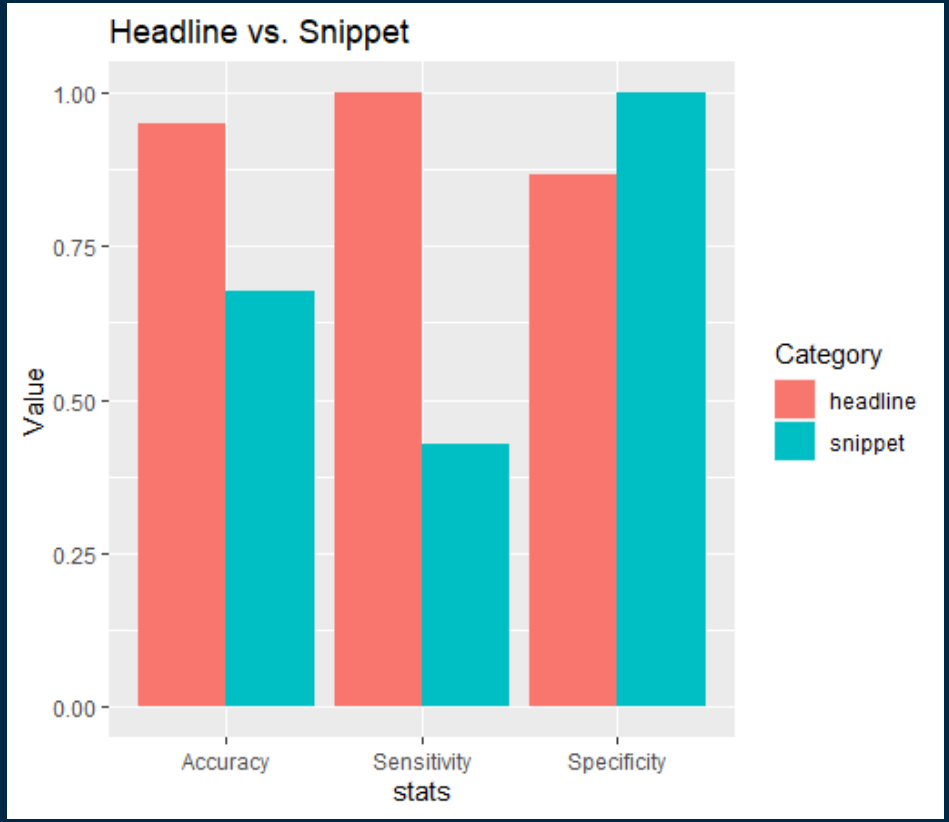
Using snippet on the left, we get the accuracy 0.6667,
Sensitivity:0.4286
Specificity: 1.000

Using headline classifier on the right, we get
accuracy :0.95,
sensitivity: 1.0000,
specificity: 0.667

Clearly using headline classifier is better

SEARCH TRUMP IN NYT CLASSIFIER

We can see that using headline classifier vs. Snippet classifier, very clearly both accuracy and sensitivity is significantly higher while snippet classifier has higher specificity



TAKE AWAY

-Having some question in understanding how Navie bays' equation on paper.

$$P(D|+) = \frac{P(+|D)P(D)}{P(+)}$$
$$= \frac{P(+|D)P(D)}{P(+|D)P(D) + P(+|N)P(N)}$$

How did we get P(+) change to P(+|D)P(D)+P(+|N)P(N)?

-Running into problem when using NYT snippet classifier, having a hard time re-creating that classifier