**FILIPPO RIVA**

**MT3100 (Brandt)**
**Statistical Learning**

**Assignment 10**

**In our final assignment, we will do some training and testing for a regression model, and we will look a bit more closely at Naïve Bayes classification.**

**1. We began the semester by looking at a linear model for life expectancy. As we have done in recently classification assignments, we will do some training and testing. There are 183 countries. We will use 143 (78%) countries to train our models and test them on the remaining 40 (22%) countries.**

**a. Using the entire data set (I named it LE), find a linear model for MaleLE as a function of FemaleLE (MaleLE and FemaleLE are the column headings in the data set). Call this model myMod1. Include the summary command for your model. You should see 181 degrees of freedom and RSE of 1.958.**

**CODE:**

```
lifexp = read.csv(file="LifeExpectancy.csv",head=TRUE,sep=",",stringsAsFactors = FALSE)

Female=lifexp$FemaleLE
Male=lifexp$MaleLE
Country=lifexp$country
myMod1=lm(Male~Female)
summary (myMod1)
```

```
Call:
lm(formula = Male ~ Female)

Residuals:
    Min      1Q  Median      3Q     Max
-6.4655 -0.9061  0.3280  1.2227  3.8673

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.07589    1.28717    2.39   0.0179 *
Female       0.89239    0.01735   51.45   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.958 on 181 degrees of freedom
Multiple R-squared:  0.936,     Adjusted R-squared:  0.9356
F-statistic:  2647 on 1 and 181 DF,  p-value: < 2.2e-16
```

**b. Now we will randomly scramble the entire data set and train on the first 143 entries. I used these commands to set things up (they should look familiar):**

```
set.seed(2)
scram=sample.int(183,183)
newLE=LE[scram,] # Same set with rows scrambled.

 trn=rep(TRUE,143) # Same setup as before.
tst=rep(FALSE,40)
train=c(trn,tst)

newLE.train=newLE[train,] # First 143 rows of newLE
newLE.test=newLE[!train,] # Last 40 rows of newLE
```

**Now you have newLE.train, which is a data set with 143 rows. Find a model, myMod2, for MaleLE as a function of FemaleLE using your training data. I used these commands to create and review the model.**

```
myMod2=lm(MaleLE~FemaleLE,data=newLE.train)
summary(myMod2)
```

**The coefficients, RSE, etc. of your model will depend on the seed, but you should see 141 degrees of freedom since you are using only the training data. Now we can make predictions on the test set.**

```
myMod2.pred=predict(myMod2,newLE.test)
```

**The object myMod2.pred is a list of predicted male life expectancies for the test countries. The actual male life expectancies for these countries are in the list newLE.test$MaleLE. Finally, we can compare our predictions to the actual values using the root mean squared error (the square root of the average of the squares of the errors):**

```
library(Metrics)
 rmse(myMod2.pred,newLE.test$MaleLE)
```

The rmse() command works a bit like the table command for classification. Instead of counting the hits and misses, the command finds the square root of the average of the squares of the gaps between the predicted values and the actual values. This gives us an overall measure of how well our model performs on the test data.

Complete the table for myMod2:

**CODE:**

**Seed (10)**

```
LE=read.csv(file="LifeExpectancy.csv",head=TRUE,sep=",",stringsAsFactors = FALSE)
```

```
Female=LE$FemaleLE
Male=LE$MaleLE
Country=LE$country
myMod1=lm(Male~Female)

set.seed(10)

scram=sample.int(183,183)
newLE=LE[scram,] # Same set with rows scrambled.

trn=rep(TRUE,143) # Same setup as before.
tst=rep(FALSE,40)
train=c(trn,tst)

newLE.train=newLE[train,] # First 143 rows of newLE
newLE.test=newLE[!train,] # Last 40 rows of newLE

myMod2=lm(MaleLE~FemaleLE,data=newLE.train)
summary(myMod2)
myMod2.pred=predict(myMod2,newLE.test)
FCITS = newLE.test[1,] #First country in the test set
library(Metrics)
print(FCITS)
rmse(myMod2.pred,newLE.test$MaleLE)
```

**Seed (20)**

```
LE=read.csv(file="LifeExpectancy.csv",head=TRUE,sep=",",stringsAsFactors = FALSE)

Female=LE$FemaleLE
Male=LE$MaleLE
Country=LE$country
myMod1=lm(Male~Female)

set.seed(20)

scram=sample.int(183,183)
newLE=LE[scram,] # Same set with rows scrambled.

trn=rep(TRUE,143) # Same setup as before.
tst=rep(FALSE,40)
train=c(trn,tst)

newLE.train=newLE[train,] # First 143 rows of newLE
newLE.test=newLE[!train,] # Last 40 rows of newLE

myMod2=lm(MaleLE~FemaleLE,data=newLE.train)
summary(myMod2)
```

```
myMod2.pred=predict(myMod2,newLE.test)
FCITS = newLE.test[1,] #First country in the test set
library(Metrics)
print(FCITS)
rmse(myMod2.pred,newLE.test$MaleLE)
```

**Seed (30)**

```
LE=read.csv(file="LifeExpectancy.csv",head=TRUE,sep=",",stringsAsFactors = FALSE)

Female=LE$FemaleLE
Male=LE$MaleLE
Country=LE$country
myMod1=lm(Male~Female)

set.seed(30)

scram=sample.int(183,183)
newLE=LE[scram,] # Same set with rows scrambled.

trn=rep(TRUE,143) # Same setup as before.
tst=rep(FALSE,40)
train=c(trn,tst)

newLE.train=newLE[train,] # First 143 rows of newLE
newLE.test=newLE[!train,] # Last 40 rows of newLE

myMod2=lm(MaleLE~FemaleLE,data=newLE.train)
summary(myMod2)
myMod2.pred=predict(myMod2,newLE.test)
FCITS = newLE.test[1,] #First country in the test set
library(Metrics)
print(FCITS)
rmse(myMod2.pred,newLE.test$MaleLE)
```

| Seed | First country in test set | RMSE |
|---|---|---|
| 10 | Ecuador | 1.98319657817853 |
| 20 | Sweden | 1.69605644694722 |
| 30 | Guatemala | 2.31353518514889 |

**Are you comfortable with the following statement? On average, it seems that our models can predict male life expectancy within 2 years of the actual values**.

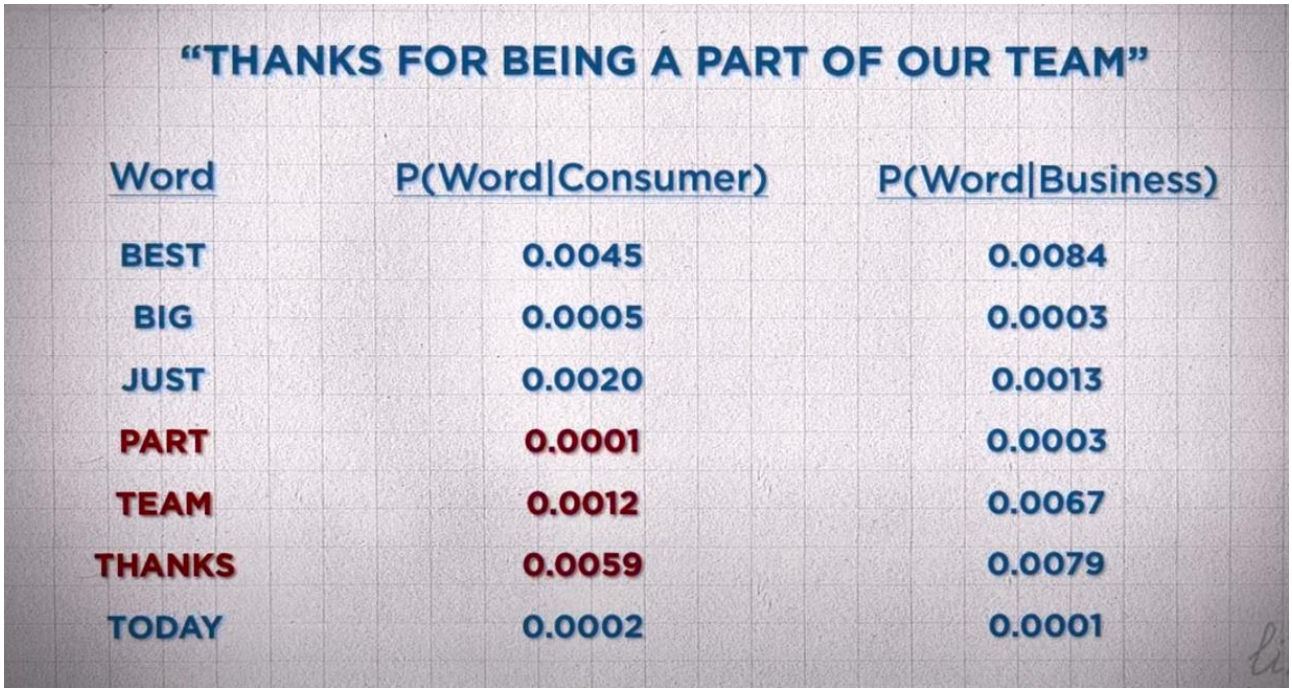
I am comfortable with stating that on average our model can predict the male life expectancy within 2 years of the actual value. By looking at the complete table above and the value of RMSE they all are really close to 2 especially if we compute the mean between the three values it is possible to find a really close number to 2.

**2. Please view the videos of Dr. Adler and Dr. Williams posted with this assignment. Then answer the following questions.**

**a. Dr. Adler used a naïve Bayes classifier to help determine which orders were made by businesses and which were made by consumers. What were the predictors in his model? Are they quantitative or qualitative? It may help to review pages 155-156 in the text.**
Dr. Alder used some important words present in messages as predictors to classify the importance of documents.
Those predictors are qualitative because they represent concepts, words instead of having numerical values.
**b. Dr. Williams worked through some of the probability and algebra behind the naïve Bayes classifier (I hope some of it looked familiar). The naïve Bayes assumption is that the predictors are independent. Explain exactly what this means in terms of the predictors that were being used. Is this assumption completely accurate in this situation? Explain.**
The assumption of independence in the context of naïve Bayes classification means that the presence or absence of each predictor (in this case the words "part", "Team" and "Thanks") is assumed to be independent of the presence or absence of other predictors when determining the importance of the document.
Example the presence of the word "Part" has nothing to do with the presence of the word "Team" in a message. So, in other words, the model assumes that the occurrence of each word is not influenced by the occurrence of any other word in the document.
In real life this assumption is not always true, there might be scenario in which certain words might be correlated with each other. An example that comes to my mind is a document containing teamwork. The words "Part" and "Team" might occur together. However, in the situation explained by Dr. Williams the assumption simplifies her computation to get strong results.

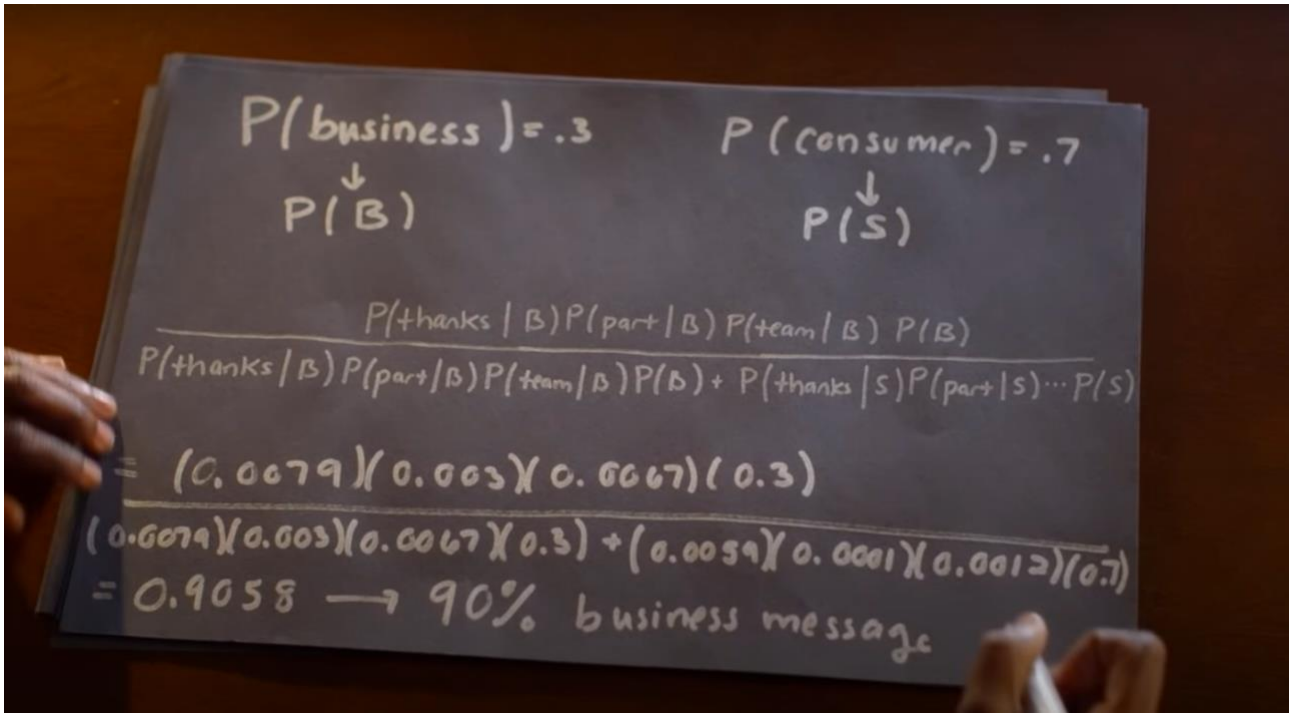
**c. When events are independent, it is very easy to compute probabilities. Complete the statement: If events A and B are independent, then P (A $\cap$ B)** = P(A) * P(B)

**d. At time 8:45 into Dr. Williams's video, she plugs in values to estimate the probability that the message comes from a business. Explain where these values come from.**




The first column represents the words used to understand the importance of the document, The second column represents the probability to find that word in a Consumer document while the third column represent the probability to find that word in A business document.

\




The probability that is a business message is P (business) = 0.3 while the probability that it is a consumer message P(consumer) is 0.7.

The formula is using the values described in the table above to understand which is the probability of the message of being a business message when the word "thanks", "part" and "team" are used in it.

In the numerator Dr. Williams multiply Probability of “thanks” in a business message, Probability of “part” in a business message, probability of “team” in a business message and total probability for business message.
In the denominator Dr. Williams add the numerator to the probability of “thanks” in a consumer message times the probability of “part” in a consumer message times the probability of “team” in a consumer message times the total probability of consumer message.