# Assignment 5: Regression and Classification

Tayyab Munir - 11716089

10/29/2020

# Question 1

This question involves the use of multiple linear regression on the Auto data set from the course webpage (https://scads.eecs.wsu.edu/index.php/datasets/ (https://scads.eecs.wsu.edu/index.php/datasets/)). Ensure that you remove missing values from the dataframe, and that values are represented in the appropriate types.

```
Auto=read.csv("https://scads.eecs.wsu.edu/wp-content/uploads/2017/09/Auto.csv", header=TRUE, na.
strings = "?")
Auto = na.omit(Auto)
head(Auto)
```

| … | cylinders | displacement | horsepower | wei… | acceleration | y… | origin | name |
|---|---|---|---|---|---|---|---|---|
| <dbl> | <int> | <dbl> | <int> | <int> | <dbl> | <int> | <int> | <chr> |
| 1 18 | 8 | 307 | 130 | 3504 | 12.0 | 70 | 1 | chevrolet chevelle m |
| 2 15 | 8 | 350 | 165 | 3693 | 11.5 | 70 | 1 | buick skylark 320 |
| 3 18 | 8 | 318 | 150 | 3436 | 11.0 | 70 | 1 | plymouth satellite |
| 4 16 | 8 | 304 | 150 | 3433 | 12.0 | 70 | 1 | amc rebel sst |
| 5 17 | 8 | 302 | 140 | 3449 | 10.5 | 70 | 1 | ford torino |
| 6 15 | 8 | 429 | 198 | 4341 | 10.0 | 70 | 1 | ford galaxie 500 |

6 rows

a. (5%) Perform a multiple linear regression with mpg as the response and all other variables except name as the predictors. Show a printout of the result (including coefficient, error and t values for each predictor). Comment on the output:

```
Auto1 = Auto[1:8]
lm.fit = lm(mpg~., data = Auto1)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = Auto1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## cylinders     -0.493376   0.323282  -1.526  0.12780
## displacement   0.019896   0.007515   2.647  0.00844 **
## horsepower    -0.016951   0.013787  -1.230  0.21963
## weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## acceleration   0.080576   0.098845   0.815  0.41548
## year           0.750773   0.050973  14.729  < 2e-16 ***
## origin         1.426141   0.278136   5.127 4.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.328 on 384 degrees of freedom
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

i. Which predictors appear to have a statistically significant relationship to the response, and how do you determine this?

Answer:

The F-statistic (252.4) shows that at least one of the variables is significant and, the p-values of variables shows that displacement, weight, year and origin are significant variables.

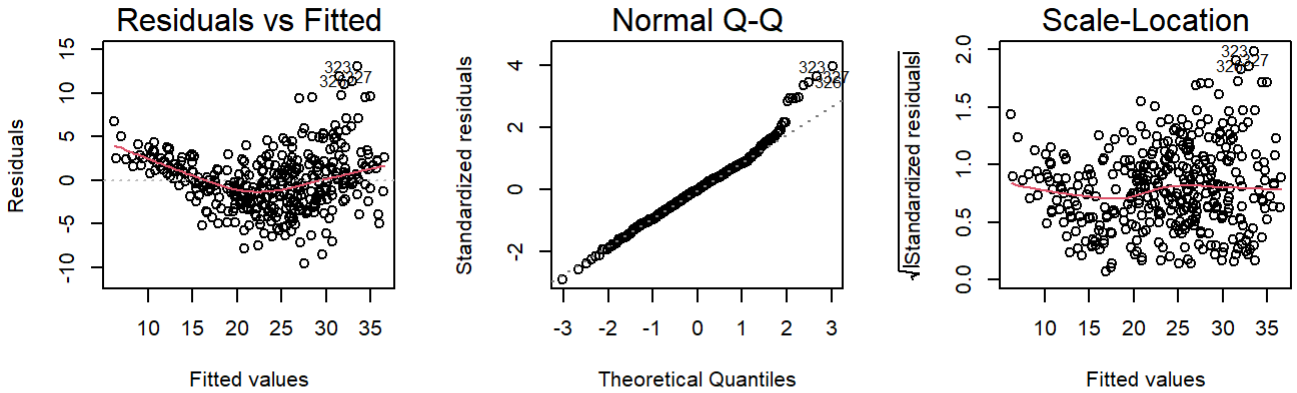
ii. What does the coefficient for the displacement variable suggest, in simple terms?

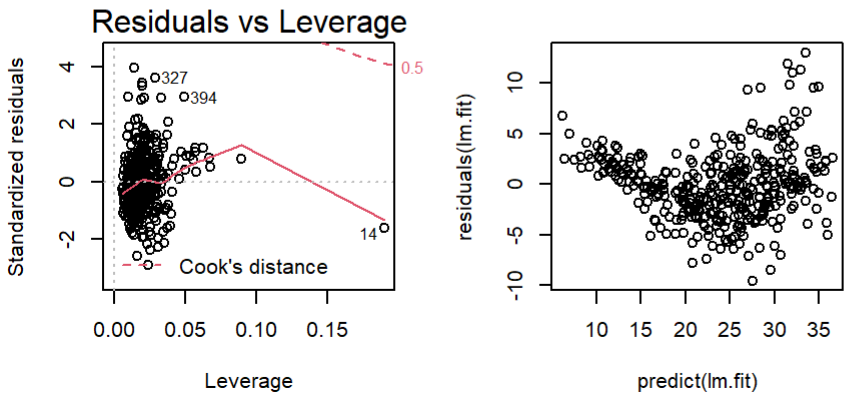
Answer:

Coefficient for displacement suggests that mpg has a positive relation with displacement. For every increase in displacement, the mpg increases by 0.019896 times and, the p-value shows that displacement is significant in predicting mpg as response variable.

b. (5%) Produce diagnostic plots of the linear regression fit. Comment on any problems you see with the fit. Do the residual plots suggest any unusually large outliers? Does the leverage plot identify any observations with unusually high leverage?

```
par(mfrow=c(2,3))
plot(lm.fit)
plot(predict(lm.fit), residuals(lm.fit))
```




# Residuals vs Fitted

The plot above shows that there is a non linear trend in the residuals vs fitted. Non-linear relationship is not considered by the linear regression. The residuals increase as the fitted values increase because there are more outliers at higher fitted values.

# Normal Q-Q Plot

For a long range of fitted values, the residuals are normally distributed but at higher range of fitted values, residual don't have normal distribution and deviate from straight line

# Residuals vs Leverage plot

Residual vs Leverage plot shows that majority of residuals are within the dotted redline called cook's distance. Here probably excluding 14th observation can enhance accuracy of linear regression.

c. (5%) Fit linear regression models with interaction effects. Do any interactions appear to be statistically significant?

```
Auto1 = Auto[1:8]
lm.fit=lm(mpg~.*.,data=Auto1)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = mpg ~ . * ., data = Auto1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.6303 -1.4481  0.0596  1.2739 11.1386
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 3.548e+01  5.314e+01   0.668  0.50475
## cylinders                   6.989e+00  8.248e+00   0.847  0.39738
## displacement               -4.785e-01  1.894e-01  -2.527  0.01192 *
## horsepower                  5.034e-01  3.470e-01   1.451  0.14769
## weight                      4.133e-03  1.759e-02   0.235  0.81442
## acceleration               -5.859e+00  2.174e+00  -2.696  0.00735 **
## year                        6.974e-01  6.097e-01   1.144  0.25340
## origin                     -2.090e+01  7.097e+00  -2.944  0.00345 **
## cylinders:displacement     -3.383e-03  6.455e-03  -0.524  0.60051
## cylinders:horsepower        1.161e-02  2.420e-02   0.480  0.63157
## cylinders:weight            3.575e-04  8.955e-04   0.399  0.69000
## cylinders:acceleration      2.779e-01  1.664e-01   1.670  0.09584 .
## cylinders:year             -1.741e-01  9.714e-02  -1.793  0.07389 .
## cylinders:origin            4.022e-01  4.926e-01   0.816  0.41482
## displacement:horsepower    -8.491e-05  2.885e-04  -0.294  0.76867
## displacement:weight         2.472e-05  1.470e-05   1.682  0.09342 .
## displacement:acceleration -3.479e-03  3.342e-03  -1.041  0.29853
## displacement:year           5.934e-03  2.391e-03   2.482  0.01352 *
## displacement:origin         2.398e-02  1.947e-02   1.232  0.21875
## horsepower:weight          -1.968e-05  2.924e-05  -0.673  0.50124
## horsepower:acceleration    -7.213e-03  3.719e-03  -1.939  0.05325 .
## horsepower:year            -5.838e-03  3.938e-03  -1.482  0.13916
## horsepower:origin           2.233e-03  2.930e-02   0.076  0.93931
## weight:acceleration         2.346e-04  2.289e-04   1.025  0.30596
## weight:year                -2.245e-04  2.127e-04  -1.056  0.29182
## weight:origin              -5.789e-04  1.591e-03  -0.364  0.71623
## acceleration:year           5.562e-02  2.558e-02   2.174  0.03033 *
## acceleration:origin         4.583e-01  1.567e-01   2.926  0.00365 **
## year:origin                 1.393e-01  7.399e-02   1.882  0.06062 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.695 on 363 degrees of freedom
## Multiple R-squared:  0.8893, Adjusted R-squared:  0.8808
## F-statistic: 104.2 on 28 and 363 DF,  p-value: < 2.2e-16
```

acceleration:origin is statistically significant because it has a low p-value of 0.00365.

acceleration:year is statistically significant because it has a low p-value of 0.03033.

displacement:year is statistically significant because it has a low p-value of 0.01352.

# Question 2

This problem involves the Boston data set, which we saw in class. We will now try to predict per capita crime rate using the other variables in this data set. In other words, per capita crime rate is the response, and the other variables are the predictors.

```
library(MASS)
names(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

a. (6%) For each predictor, fit a simple linear regression model to predict the response. Include the code, but not the output for all models in your solution.

```
coeff_Uni=0

lm.fit=lm(crim~zn,data=Boston)
summary(lm.fit)
coeff_U=coef(lm.fit)
coeff_Uni[1]=coeff_U[2]

lm.fit=lm(crim~indus,data=Boston)
summary(lm.fit)
coeff_U=coef(lm.fit)
coeff_Uni[2]=coeff_U[2]

lm.fit=lm(crim~chas,data=Boston)
summary(lm.fit)
coeff_U=coef(lm.fit)
coeff_Uni[3]=coeff_U[2]

lm.fit=lm(crim~nox,data=Boston)
summary(lm.fit)
coeff_U=coef(lm.fit)
coeff_Uni[4]=coeff_U[2]

lm.fit=lm(crim~rm,data=Boston)
summary(lm.fit)
coeff_U=coef(lm.fit)
coeff_Uni[5]=coeff_U[2]

lm.fit=lm(crim~age,data=Boston)
summary(lm.fit)
coeff_U=coef(lm.fit)
coeff_Uni[6]=coeff_U[2]

lm.fit=lm(crim~dis,data=Boston)
summary(lm.fit)
coeff_U=coef(lm.fit)
coeff_Uni[7]=coeff_U[2]

lm.fit=lm(crim~rad,data=Boston)
summary(lm.fit)
coeff_U=coef(lm.fit)
coeff_Uni[8]=coeff_U[2]

lm.fit=lm(crim~tax,data=Boston)
summary(lm.fit)
coeff_U=coef(lm.fit)
coeff_Uni[9]=coeff_U[2]

lm.fit=lm(crim~ptratio,data=Boston)
summary(lm.fit)
coeff_U=coef(lm.fit)
coeff_Uni[10]=coeff_U[2]

lm.fit=lm(crim~black,data=Boston)
```

```
summary(lm.fit)
coeff_U=coef(lm.fit)
coeff_Uni[11]=coeff_U[2]

lm.fit=lm(crim~lstat,data=Boston)
summary(lm.fit)
coeff_U=coef(lm.fit)
coeff_Uni[12]=coeff_U[2]

lm.fit=lm(crim~medv,data=Boston)
summary(lm.fit)
coeff_U=coef(lm.fit)
coeff_Uni[13]=coeff_U[2]
```

b. (6%) In which of the models is there a statistically significant association between the predictor and the response? Considering the meaning of each variable, discuss the relationship between crim and nox, chas, medv and dis in particular. How do these relationships differ?

• lstat is statistically significant because of low p-value (p-val<2e-16). This shows the correlation between lower status of population and the crime rate. positive value indicates higher the lower status of population, higher will be the crime rate.

• zn is a statistically significant predictor for crime rate because of low p-value (5.51e-6). This has a negative correlation with crime rate. The bigger the residential lands are, the lower will be the crime rate.

• Indus is statistically significant because of lower p-value (p-val<2.2e-16). Positive value shows the relation between business acres per town and crime rate. The more business friendly areas, the more crimes will be there.

• chas is statistically insignificant (p-val=0.209).

• nox is statistically significant because of low p-value (p-val<2e-16). Correlation is positive between nitrogen oxide concentration and the crime rate.The more nitogen concentration, the more will be the crime.

• rm is statistically significant because of low p-value (p-val=6.35e-16). The correlation coefficient is negative. This shows, the more the population, the more will be crime.

• dis is significant because of low p-value (p-val<2e-16). The negative value shows the negative correlation.The more the distance from employment centres, the lower will be the crime rate.

• tax is statistically significant (p-val<2e-16). There is positive correlation between property tax and crime rate. The higher the property tax, higher will be the crime rate.

• rad is significant (p-val=2e-16). There is positive correlation. The more accessibility to radial highways, the more will be the crime rate.

• ptratio is statistically significant (p-val=2.94e-11). here is a positive correlation. Higher the pupilteacher ratio, higher will be the crime rate.

• black is statistically significant (p-val<2e-16). There is a negative correlation coefficient. The higher the black population, lower will be the crime rate.

• medv is statistically significant (p-val<2e-16)There is a negative correlation. The lower the median value of owner occupied home, higher will be the crime rate

c. (6%) Fit a multiple regression model to predict the response using all the predictors. Describe your results. For which predictors can we reject the null hypothesis H0 : $\beta j = 0$?

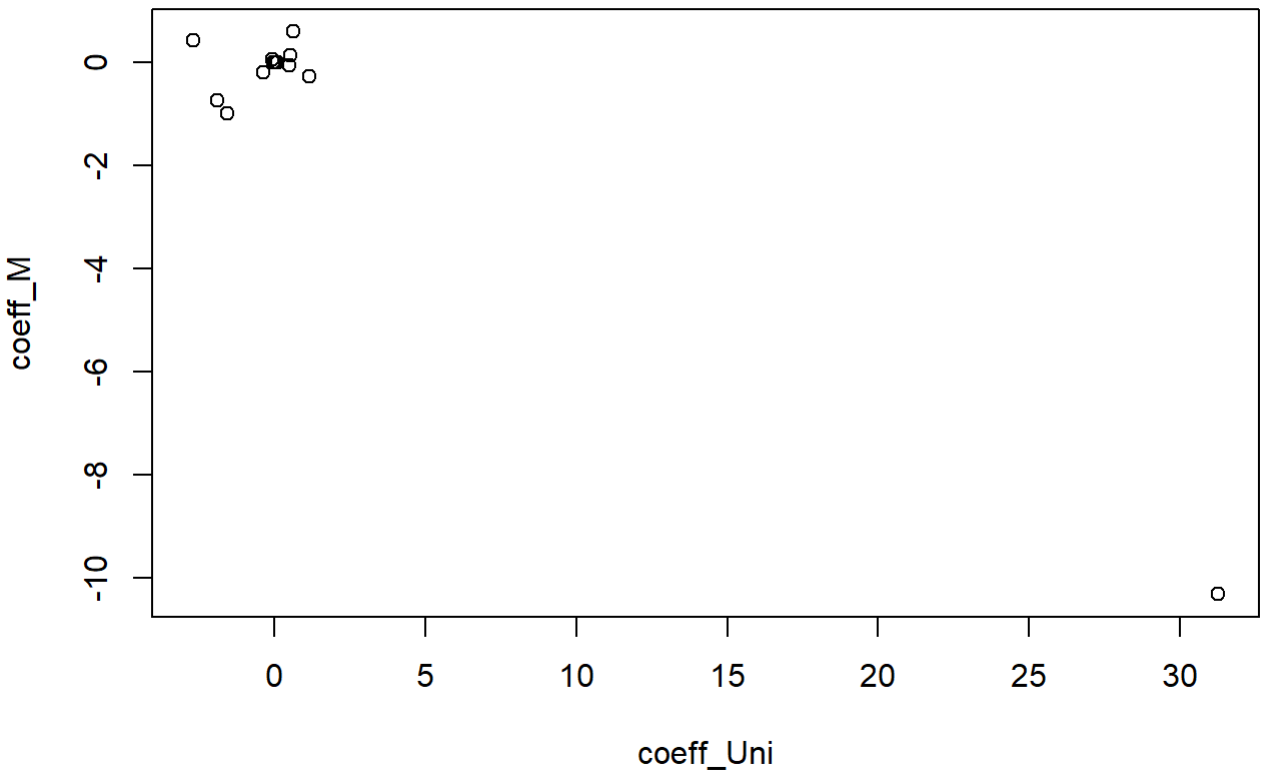
```
lm.fit=lm(crim~.,data=Boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = crim ~ ., data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.033228   7.234903   2.354 0.018949 *
## zn            0.044855   0.018734   2.394 0.017025 *
## indus        -0.063855   0.083407  -0.766 0.444294
## chas         -0.749134   1.180147  -0.635 0.525867
## nox         -10.313535   5.275536  -1.955 0.051152 .
## rm            0.430131   0.612830   0.702 0.483089
## age           0.001452   0.017925   0.081 0.935488
## dis          -0.987176   0.281817  -3.503 0.000502 ***
## rad           0.588209   0.088049   6.680 6.46e-11 ***
## tax          -0.003780   0.005156  -0.733 0.463793
## ptratio      -0.271081   0.186450  -1.454 0.146611
## black        -0.007538   0.003673  -2.052 0.040702 *
## lstat         0.126211   0.075725   1.667 0.096208 .
## medv         -0.198887   0.060516  -3.287 0.001087 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 492 degrees of freedom
## Multiple R-squared:  0.454,  Adjusted R-squared:  0.4396
## F-statistic: 31.47 on 13 and 492 DF,  p-value: < 2.2e-16
```

zn,nox,lstat,dis,rad,black,and medv are significant on the basis of p-value and null hypothesis can be rejected for them

d. (6%) How do your results from (a) compare to your results from (c)? Create a plot displaying the univariate regression coefficients from (a) on the x-axis, and the multiple regression coefficients from (c) on the y-axis. That is, each predictor is displayed as a single point in the plot. Its coefficient in a simple linear regression model is shown on the x-axis, and its coefficient estimate in the multiple linear regression model is shown on the y-axis. What does this plot tell you about the various predictors?

```
lm.fit=lm(crim~.,data=Boston)
coeff_M=coef(lm.fit)
coeff_M=coeff_M[2:14]
plot(coeff_M~coeff_Uni)
```




One of the variables in uni-variate coefficient differs greatly from multi-variate coefficient. In case of other variables, the uni-variate coefficient is close to multi-variate coefficient. The difference exists between multi-variate and co-variate estimates because the variables could be correlated.

e. (6%) Is there evidence of non-linear association between any of the predictors and the response? To answer this question, for each predictor X, fit a model of the form $Y = \beta0 + \beta1X + \beta2X2 + \beta3X3+ \varepsilon$ Hint: use the poly() function in R. Again, include the code, but not the output for each model in your solution, and instead describe any non-linear trends you uncover.

```
summary(lm(crim ~ zn + I(zn^2) + I(zn^3), data = Boston))

summary(lm(crim ~ indus + I(indus^2) + I(indus^3), data = Boston))

summary(lm(crim ~ nox + I(nox^2) + I(nox^3), data = Boston))

summary(lm(crim ~ rm + I(rm^2) + I(rm^3), data = Boston))

summary(lm(crim ~ age + I(age^2) + I(age^3), data = Boston))

summary(lm(crim ~ dis + I(dis^2) + I(dis^3), data = Boston))

summary(lm(crim ~ rad + I(rad^2) + I(rad^3), data = Boston))

summary(lm(crim ~ tax + I(tax^2) + I(tax^3), data = Boston))

summary(lm(crim ~ ptratio + I(ptratio^2) + I(ptratio^3), data = Boston))

summary(lm(crim ~ black + I(black^2) + I(black^3), data = Boston))

summary(lm(crim ~ lstat + I(lstat^2) + I(lstat^3), data = Boston))

summary(lm(crim ~ medv + I(medv^2) + I(medv^3), data = Boston))
```

There is a non linear relationship for Nox, ptratio, dis, and medv with the response variable. This is clear from p-values for squared and cube terms of the prediction variables and, there is a non linear relationship for the age because only the cube and square values of age have p-value which are significant. Furthermore, no evidence of non linear relation for the rest of them.

# Question 3

Suppose we collect data for a group of students in a statistics class with variables:

X1 = hours studied,

X2 = undergrad GPA,

X3 = PSQI score (a sleep quality index), and

Y = receive an A.

We fit a logistic regression and produce estimated coefficient, β0 = −7, β1 = 0.1, β2 = 1, β3 = -.04.

a. (5%) Estimate the probability that a student who studies for 32 h, has a PSQI score of 12 and has an undergrad GPA of 3.0 gets an A in the class. Show your work.

Solution:

probability= eB0+B1X1+B2X2+B3X3/1+eB0+B1X1+B2X2+B3X3 = A/B

```
A = 2.71828^(-7+(0.1)*(32)+(1)*(3)+(-0.04)*(12))
B = 1+2.71828^(-7+(0.1)*(32)+(1)*(3)+(-0.04)*(12))
probability = A/B
probability
```

```
## [1] 0.2175504
```

b. (5%) How many hours would the student in part (a) need to study to have a 50 % chance of getting an A in the class? Show your work.

Solution:

log ((p(x)/1-p(x)) = BO+B1X1+B2X2+B3X3

```
C = log((0.5)/(1-0.5))
study_hours = (7-(1)*(3)+(0.04)*(12))/0.1
study_hours
```

```
## [1] 44.8
```

c. (5%) How many hours would a student with a 3.0 GPA and a PSQI score of 3 need to study to have a 50 % chance of getting an A in the class? Show your work.

Solution:

log ((p(x)/1-p(x)) = BO+B1X1+B2X2+B3X3

```
D = log((0.5)/(1-0.5))
hours_for_A = (7-(1)*(3)+(0.04)*(3))/0.1
hours_for_A
```

```
## [1] 41.2
```

# Question 4

For this question, you will a naïve Bayes model to classify newspaper articles by their section. You will be provided a set of news articles (http://scads.eecs.wsu.edu/index.php/datasets (http://scads.eecs.wsu.edu/index.php/datasets)) collected from the Guardian (a British newspaper). The articles are cleared of major confounding factors, such as HTML tags, but it is up to you to check the articles for other problems and to prepare them for classification.

```
articles = read.csv("C:/Users/tayya/OneDrive/Desktop/Fall 2020/Data Science/Assignment 5/Data/Gu
ardianArticles.csv")


func_to_clean = function(stringData) {
  stringData = gsub("<.*?>", "", stringData)
  stringData = gsub("[[:punct:]]", "", stringData)
  stringData = gsub("[[:digit:]]", "", stringData)
  stringData = tolower(stringData)
  return(stringData)
}

articles1 = func_to_clean(articles)
```

a. Tokenization (20%) In order to use Naïve Bayes effectively, you will need to split your text into tokens. It is common practice when doing this to reduce your words to their stems so that conjugations produce less noise in your data. For example, the words “speak”, “spoke”, and “speaking” are all likely to denote a similar context, and so a stemmed tokenization will merge all of them into a single stem. R has several libraries for tokenization, stemming and text mining. Some you may want to use as a starting point are tokenizers, SnowballC, tm respectively, or alternatively quanteda, which will handle the aforementioned along with building your model in the next step. You will need to produce a document-term matrix from your stemmed tokenized data. This will have a very wide feature set (to be reduced in the following step) where each word stem is a feature, and each article has a list of values representing the number of occurrences of each stem in its body. Before representing the feature set in a non-compact storage format (such as a plain matrix), you will want to remove any word which appears in too few documents (typically fewer than 1% of documents, but you can be more or less stringent as you see fit). You may also use a boolean for word presence/absence if you find it more effective. To demonstrate your completion of this part, you can simply select and print the text of a random article along with the non-zero entries of its feature vector.

```
library(tm)
```

```
## Loading required package: NLP
```

```
library(tokenizers)
```

```
## Warning: package 'tokenizers' was built under R version 4.0.3
```

```
library(corpus)
```

```
## Warning: package 'corpus' was built under R version 4.0.3
```

```
body = Corpus(VectorSource(articles$body))

term_doc_matrix = TermDocumentMatrix(body)

as.matrix(term_doc_matrix[17, which(as.matrix(term_doc_matrix[17, ]) != 0)])
```

```
##                Docs
## Terms           1 148 218 223 239 254 261 277 305 306 308 326 346 352 363 381
##   administration 1   1   1   1   1   3   1   1   4   1   1   1   1   1   1   1
##                Docs
## Terms           441 503 505 535 574 580 582 642 667 786 811 812 834 945 956
##   administration   2   2   1   2   1   1   1   1   1   2   2   1   1   1   1
##                Docs
## Terms           970 972 989 990 998 1011 1018 1025 1034 1050 1060 1067 1070
##   administration   1   1   2   2   3    1    1    1    1    1    3    1    1
##                Docs
## Terms           1117 1118 1126 1128 1134 1138 1147 1150 1157 1206 1214 1227
##   administration    1    1    1    1    1    1    2    1    1    2    1    1
##                Docs
## Terms           1315 1330 1342 1350 1364 1365 1369 1386 1417 1421 1428 1435
##   administration    1    1    1    1    1    1    1    1    2    1    4    1
##                Docs
## Terms           1437 1452 1465 1491 1492 1501 1515 1541 1544 1545 1546 1552
##   administration    1    3    1    1    1    4    1    2    1    1    5    1
##                Docs
## Terms           1554 1556 1562 1574 1586 1601 1602 1617 1625 1626 1629 1639
##   administration    2    2    6    2    1    1    1    1    4    2    3    2
##                Docs
## Terms           1641 1663 1672 1682 1696 1706 1709 1713 1730 1733 1814 1826
##   administration    1    1    2    1    2    1    1    1    1    1    1    1
##                Docs
## Terms           1942 1949 1961 1985 2002 2046 2097 2115 2125 2133 2135 2136
##   administration    1    1    1    1    2    1    1    1    1    1    1    1
##                Docs
## Terms           2137 2138 2146 2151 2168 2181 2193 2202 2223 2228 2230 2251
##   administration    1    1    1    1    3    1    4    1    1    1    1    1
##                Docs
## Terms           2294 2297 2300 2410 2417 2446 2510 2527 2528 2539 2571 2574
##   administration    1    1    4    2    1    1    5    1    3    2    1    1
##                Docs
## Terms           2577 2580 2593 2594 2596 2598 2600 2606 2611 2623 2626 2647
##   administration    1    2    1    2    1    4    2    1    8    2    7    6
##                Docs
## Terms           2651 2652 2654 2658 2668 2683 2694 2696 2720 2735 2770 2774
##   administration    4    1    4    3    2    5    1    1    1    1    4    1
##                Docs
## Terms           2780 2798 2821 2851 2862 2866 3069 3097 3115 3119 3149 3240
##   administration    1    1    1    1    1    2    1    2    1    1    1    2
##                Docs
## Terms           3258 3264 3275 3278 3287 3291 3295 3311 3332 3372 3384 3391
##   administration    1    1    1    1    1    1    1    1    1    4    1    1
##                Docs
## Terms           3394 3411 3418 3420 3421 3428 3432 3444 3500 3501 3583 3650
##   administration    1    2    1    2    1    1    1    1    1    1    1    1
##                Docs
## Terms           3655 3662 3674 3678 3684 3686 3691 3701 3716 3722 3729 3766
##   administration    1    1    1    3    2    2    2    1    1    1    3    1
##                Docs
## Terms           3767 3769 3778 3792 3795 3813 3819 3829 3830 3882 3886 3891
```

```
##   administration    1    1    3    4    1    3    1    2    1    1    2    1
##                 Docs
## Terms            3901 3933 3951 3955 3964 3974 3988 4001 4013 4127 4140 4183
##   administration    1    1    1    1    2    1    1    1    1    1    1    1
##                 Docs
## Terms            4233 4278 4288 4291 4389 4395 4400 4406 4411 4419 4421 4424
##   administration    1    1    3    1    3    2    1    1    1    1    1    1
##                 Docs
## Terms            4432 4470 4478 4487 4500 4515 4526 4553 4557 4562 4570 4576
##   administration    3    1    2    1    4    1    1    1    1    1    1    1
##                 Docs
## Terms            4581 4608 4698 4719 4743 4770 4774 4789 4792 4800 4803 4812
##   administration    3    1    1    1    1    2    1    1    1    2    3    1
##                 Docs
## Terms            4814 4827 4828 4829 4860 4870 4873 4895 4926 4938 4941 4943
##   administration    2    3    1    1    2    1    1    2    1    1    1    1
##                 Docs
## Terms            4960 4963 4970 4991 5004 5008 5018 5036 5064 5069 5079 5090
##   administration    1    1    2    2    1    1    2    2    2    2    1    2
##                 Docs
## Terms            5096 5097 5102 5108 5116 5120 5121 5125 5160 5233 5234 5325
##   administration    1    2    1    2    1    3    1    1    1    1    2    1
##                 Docs
## Terms            5450 5488 5494 5518 5530 5532 5542 5548 5556 5564 5572 5574
##   administration    1    1    1    1    3    3    2    1    1    3    2    1
##                 Docs
## Terms            5577 5579 5589 5629 5661 5676 5681 5702 5716 5819 5930 5950
##   administration    1    2    1    1    1    1    1    1    1    1    3    1
##                 Docs
## Terms            5955 5960 5972 5979 6009 6023 6032 6035 6039 6044 6053 6063
##   administration    1    1    1    1    1    1    7    1    1    7    4    2
##                 Docs
## Terms            6075 6077 6081 6087 6097 6102 6120 6133 6143 6152 6157 6169
##   administration    1    1    1    1    1    1    3    2    1    1    2    1
##                 Docs
## Terms            6171 6175 6185 6194 6203 6207 6209 6214 6222 6276 6287 6345
##   administration    1    1    2    1    1    1    1    1    1    1    1    1
##                 Docs
## Terms            6422 6468 6495 6517 6540 6567 6570 6606 6624 6633 6669 6683
##   administration    1    1    1    1    1    2    1    1    1    1    1    3
##                 Docs
## Terms            6693 6709 6714 6724 6728 6751 6755 6762 6775 6781 6793 6819
##   administration    6    1    1    1    1    1    2    1    1    2    1    2
##                 Docs
## Terms            6834 6853 6861 6871 6877 6927 7019 7037 7068 7097 7099 7105
##   administration    1    1    1    1    1    1    1    1    1    1    3    2
##                 Docs
## Terms            7106 7118 7119 7121 7162 7166 7175 7194 7195 7204 7208 7254
##   administration    2    1    2    1    1    1    3    2    3    4    1    2
##                 Docs
## Terms            7260 7274 7275 7281 7326 7353 7400 7404 7407 7412 7431 7547
##   administration    2    1    1    2    2    1    1    1    4    7    1    1
##                 Docs
## Terms            7644 7648 7650 7674 7707 7822 7838 7842 7848 7853 7854 7857
```

```
##   administration    1    1    2    1    1    1    1    3    2    9    2    2
##                 Docs
## Terms            7862 7870 7878 7879 7884 7888 7889 7891 7902 7913 7914 7930
##   administration    1    4    1    1    1    1    1    4    1    2    1    1
##                 Docs
## Terms            7933 7949 7958 7964 7966 8189 8194 8248 8251 8252 8254 8262
##   administration    2    1    2    1    1    1    2    5    1    1    1    1
##                 Docs
## Terms            8274 8275 8276 8278 8282 8288 8295 8317 8339 8360 8365 8367
##   administration    3    4    1    1    1    3    1    5    3    1    2    2
##                 Docs
## Terms            8395 8398 8417 8427 8473 8499 8687 8890 8903 8908 8939 8955
##   administration    1    1    1    1    1    1    1    1    1    1    1    1
##                 Docs
## Terms            8965 8972 9014 9045 9046 9056 9063 9071 9085 9110 9123 9142
##   administration    2    2    1    1    1    1    1    3    1    1    1    1
##                 Docs
## Terms            9144 9147 9162 9168 9269 9336 9371 9375 9407 9436 9442 9454
##   administration    1    1    1    1    1    1    1    4    1    3    3    1
##                 Docs
## Terms            9471 9474 9486 9495 9502 9512 9515 9518 9526 9529 9535 9547
##   administration    1    1    3    2    1    1    1    1    3    3    1    1
##                 Docs
## Terms            9549 9550 9556 9734 9934 9957 9970 10025 10036 10048 10122
##   administration    1    1    1    1    2    2    2     1     1     1     1
##                 Docs
## Terms            10129 10142 10146 10161 10193 10201 10221 10241 10248 10264
##   administration     1     1     1     1     1     1     2     1     1     1
##                 Docs
## Terms            10279 10281 10298 10300 10306 10312 10331 10390 10415 10527
##   administration     1     1     1     1     2     2     2     1     1     1
##                 Docs
## Terms            10533 10548 10551 10580 10581 10583 10589 10597 10598 10606
##   administration     1     2     1     1     1     4     1     1     4     1
##                 Docs
## Terms            10637 10641 10649 10673 10676 10703 10810 10888 10928 11050
##   administration     1     1     3     3     1     3     1     1     1     2
##                 Docs
## Terms            11069 11074 11076 11106 11122 11157 11190 11206 11209 11222
##   administration     1     1     1     2     1     5     1     1     2     1
##                 Docs
## Terms            11226 11228 11256 11269 11297 11301 11307 11324 11335 11352
##   administration     1     1     1     2     1     2     1     2     1     1
##                 Docs
## Terms            11387 11395 11396 11400 11423 11438 11439 11443 11447
##   administration     1     1     1     1     1     2     4     1     5
```

b. Classification (20%) For the final portion of this assignment, you will build and test a Naïve Bayes classifier with your data. First, you will need to use feature selection to reduce your feature set. A popular library for this is caret. It has many functionalities for reducing feature sets, including removing highly correlated features. You may wish to try several different methods to see which produces the best results for the following steps. Next, you will split your data into a training set and a test set. Your training set should comprise approximately 80% of your articles, however, you may try several sizes to find which produces the

best results. Whatever way you split your training and test sets, however, you should try to ensure that your six article categories are equally represented in both sets. Next, you will build your Naïve Bayes classifier from your training data. The e1071 package is most commonly used for this. Finally, you can use your model to predict the categories of your test data. Once you have produced a model that generates the best predictions you can get, print a confusion matrix of the results to demonstrate your completion of this task. For each class, give scores for precision (TruePositives / TruePositives+FalsePositives) and recall (TruePositives / TruePositives+FalseNegatives). To do this, you may want to use the confusionMatrix() function.