Project 1 Part 2: Classification

4375 Machine Learning with Dr. Mazidi

Ali Hilal

March 31, 2022

*** This notebook discusses suicide, this is just a warning in case this is a sensitive topic to read / think about ***

For the second part of this project, I am analyzing data in a data set using classification algorithms. This is the link where I got the data from, titled: WHO Suicide Statistics

https://www.kaggle.com/datasets/szamil/who-suicide-statistics

In the original data set, there are 6 columns and 43,776 rows.

##The Data Set

```
sdf <- read.csv("who_suicide_statistics.csv")
```

This is reading in the data from the file "who_suicide_statistics.csv".

##Data Cleaning:

I am applying some data cleaning techniques to the data set. First, I am renaming the columns for a cleaner look as well as easier understanding. Secondly, I am renaming the values "5-14 years" to "05-14 years". I will explain why in a second. Next, I remove all instances of N/A data so we can have complete data. I also arrange the data in descending order of year, gender and age group. This is where changing the values from "5-14 years" to "05-14 years" plays its role. Now, with the age 05 at the beginning of the string, this age group appears first in the data set, as apposed to 5th before the change. Lastly, I changed the gender, country, and age group variables to factors for easier data computation.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
names(sdf) <- c("Country", "Year", "Gender", "Age_Group", "Number_of_Suicides", "Population")
sdf[sdf == "5-14 years"] <- "05-14 years"
sdf <- sdf[complete.cases(sdf), ]
invisible(sdf %>% arrange(Year, Gender, Age_Group))
sdf$Gender <- as.factor(sdf$Gender) #Female = 1; Male = 2
sdf$Age_Group <- as.factor(sdf$Age_Group)
sdf$Country <- as.factor(sdf$Country)
```

##Data Exploration:

A small introduction to the data set we are working with. Here there is the first and last 6 elements in the data set are displayed, a summary of the data, and every column with the name and the data type.

```r
head(sdf, 6)
```

```
##    Country Year Gender   Age_Group Number_of_Suicides Population
## 25 Albania 1987 female 15-24 years                14     289700
## 26 Albania 1987 female 25-34 years                 4     257200
## 27 Albania 1987 female 35-54 years                 6     278800
## 28 Albania 1987 female 05-14 years                 0     311000
## 29 Albania 1987 female 55-74 years                 0     144600
## 30 Albania 1987 female   75+ years                 1      35600
```

```r
tail(sdf, 6)
```

```
##                    Country Year Gender   Age_Group Number_of_Suicides
## 43759 Virgin Islands (USA) 2015   male 15-24 years                 0
## 43760 Virgin Islands (USA) 2015   male 25-34 years                 2
## 43761 Virgin Islands (USA) 2015   male 35-54 years                 1
## 43762 Virgin Islands (USA) 2015   male 05-14 years                 0
## 43763 Virgin Islands (USA) 2015   male 55-74 years                 0
## 43764 Virgin Islands (USA) 2015   male   75+ years                 0
##       Population
## 43759       6933
## 43760       4609
## 43761      12516
## 43762       7291
## 43763      12615
## 43764       2496
```

```r
summary(sdf)
```

```
##         Country           Year         Gender           Age_Group
##  Hungary    :  456   Min.   :1979   female:18030   05-14 years:6010
##  Netherlands:  456   1st Qu.:1991   male  :18030   15-24 years:6010
##  Argentina  :  444   Median :2000                  25-34 years:6010
##  Austria    :  444   Mean   :1999                  35-54 years:6010
##  Belgium    :  444   3rd Qu.:2008                  55-74 years:6010
##  Brazil     :  444   Max.   :2016                  75+ years  :6010
##  (Other)    :33372
##  Number_of_Suicides   Population
##  Min.   :    0.0    Min.   :     259
```

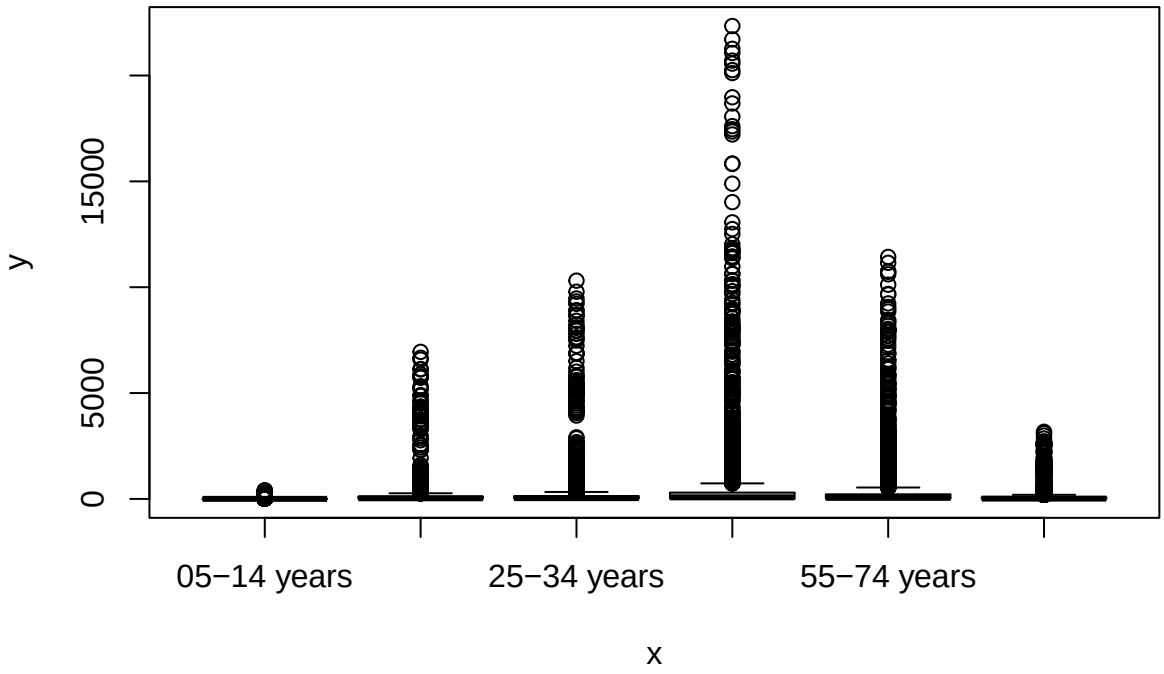
```
##  1st Qu.:    2.0    1st Qu.:   80566
##  Median :   21.0    Median :  375765
##  Mean   :  221.8    Mean   : 1699996
##  3rd Qu.:  116.0    3rd Qu.: 1344900
##  Max.   :22338.0    Max.   :43805214
##
```

```
str(sdf)
```

```
## 'data.frame':    36060 obs. of  6 variables:
##  $ Country           : Factor w/ 118 levels "Albania","Antigua and Barbuda",..: 1 1 1 1 1 1 1 1 1 1
##  $ Year              : int  1987 1987 1987 1987 1987 1987 1987 1987 1987 1987 ...
##  $ Gender            : Factor w/ 2 levels "female","male": 1 1 1 1 1 1 2 2 2 2 ...
##  $ Age_Group         : Factor w/ 6 levels "05-14 years",..: 2 3 4 1 5 6 2 3 4 1 ...
##  $ Number_of_Suicides: int  14 4 6 0 0 1 21 9 16 0 ...
##  $ Population        : int  289700 257200 278800 311000 144600 35600 312900 274300 308000 338200 ...
```

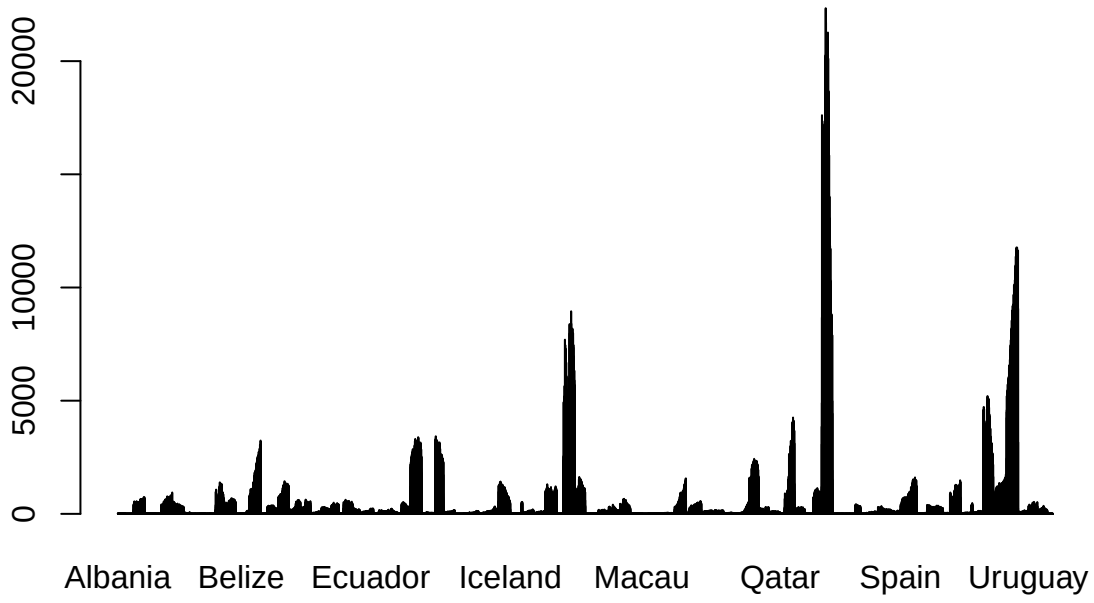
A graph of the number of suicides in each age group. The numbers increase from the age groups of 05-14 until 35-54, which sees a huge spike. Then a equally sharp decrease in the numbers in the last two groups of 55-74 and 75+.

```
plot(sdf$Age_Group, sdf$Number_of_Suicides)
```



Here is a graph with the number of suicides per country. It is hard to make out what is being displayed here, however it is really interesting to look at. It also begs the question, what instance is that extremely large spike, and where? That question is answered under this graph.

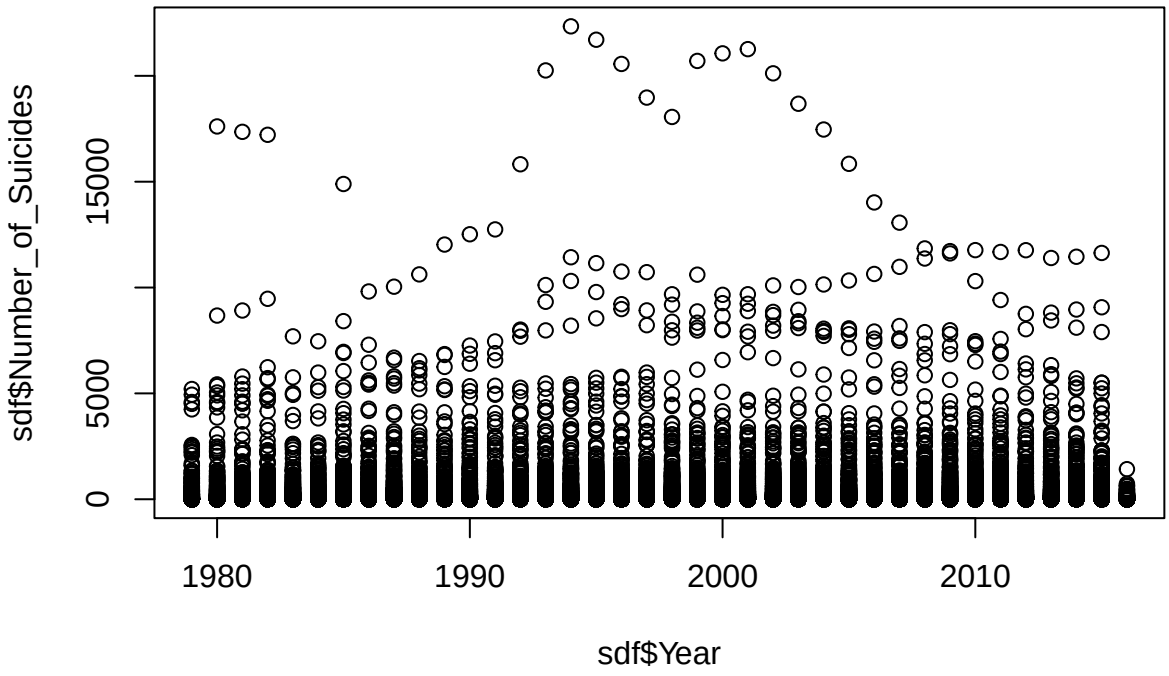
```
barplot(height=sdf$Number_of_Suicides, names=sdf$Country)
```




```
sdf[which.max(sdf$Number_of_Suicides),]
```

```
##                   Country Year Gender   Age_Group Number_of_Suicides Population
## 33129 Russian Federation 1994   male 35-54 years              22338   19044200
```

```
plot(sdf$Year, sdf$Number_of_Suicides)
```

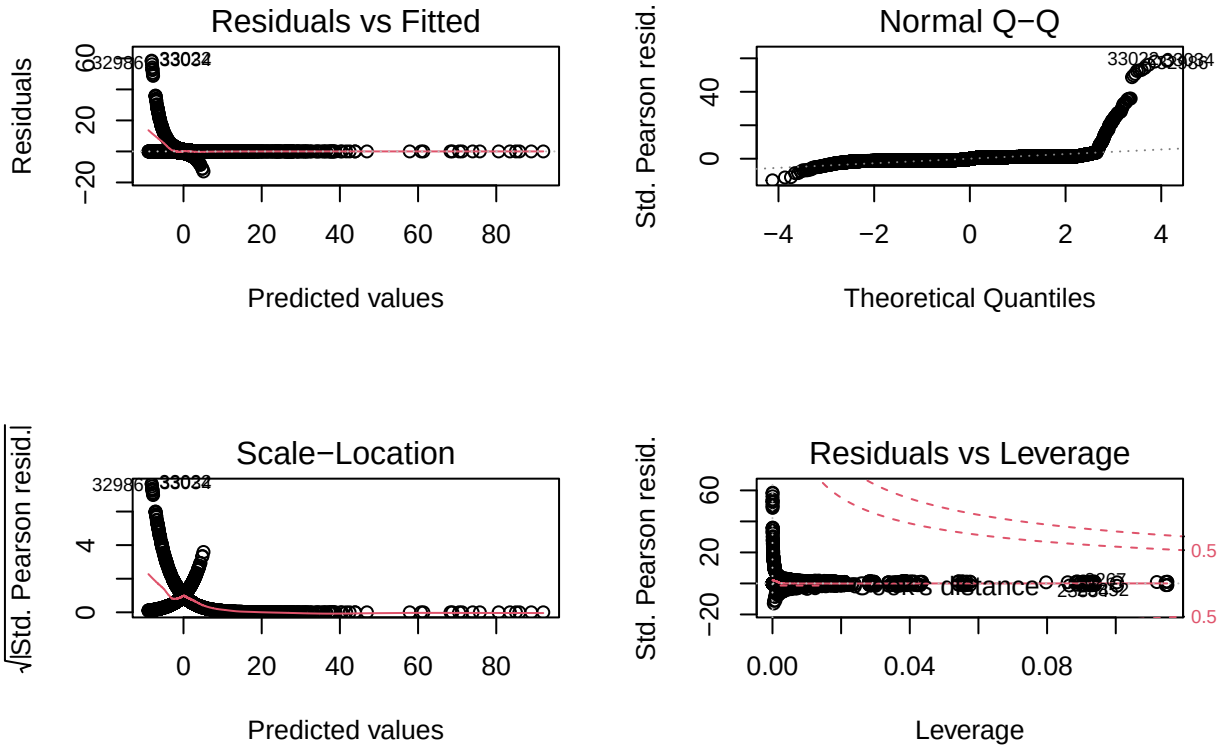


One more graph to display the number of suicides per year. It is interesting to see that there is a spike between the years 1991 and 2001.

##Logistic Regression:

With logistic regression, I created two models, a male model and a female model. When I factor the male data set, I set male as 1 and female as 0. Using that data set, I split the data into train and test and generate my male Logistic regression model.

```
set.seed(1234)
options(warn=-1)
sdfMale <- sdf
sdfMale$Gender <- as.factor(ifelse(sdfMale$Gender == "male", 1, 0))
split1 <- sample(1:nrow(sdfMale), nrow(sdfMale)*0.75, replace=FALSE)
maletrain <- sdfMale[split1, ]
maletest  <- sdfMale[-split1, ]
maleglm <- glm(Gender~.-Year, data = maletrain, family = "binomial")
par(mfrow=c(2,2))
plot(maleglm)
```



These are graphs made using the Logistic Regression Model Residuals VS Fitted: We want to see a straight, red line, and we can see that there is a small bend at the very beginning, but after that its is straight. Normal Q-Q: We want to see a straight line that sits right along the dotted line, and we see that right until the end with a large number of outliers. Scale- Location: We want to see a horizontal line with evenly spaced data on both sides, however we do not see that. Residuals VS Leverage: This graphs says what leverage points are influencing the line of regression.

```
maleprobs <- predict(maleglm, newdata = maletest)
malepred <- ifelse(maleprobs > 0.5, 1, 0)
maleacc <- mean(malepred == maletest$Gender)
print(paste("Male Accuracy: ", maleacc))
```

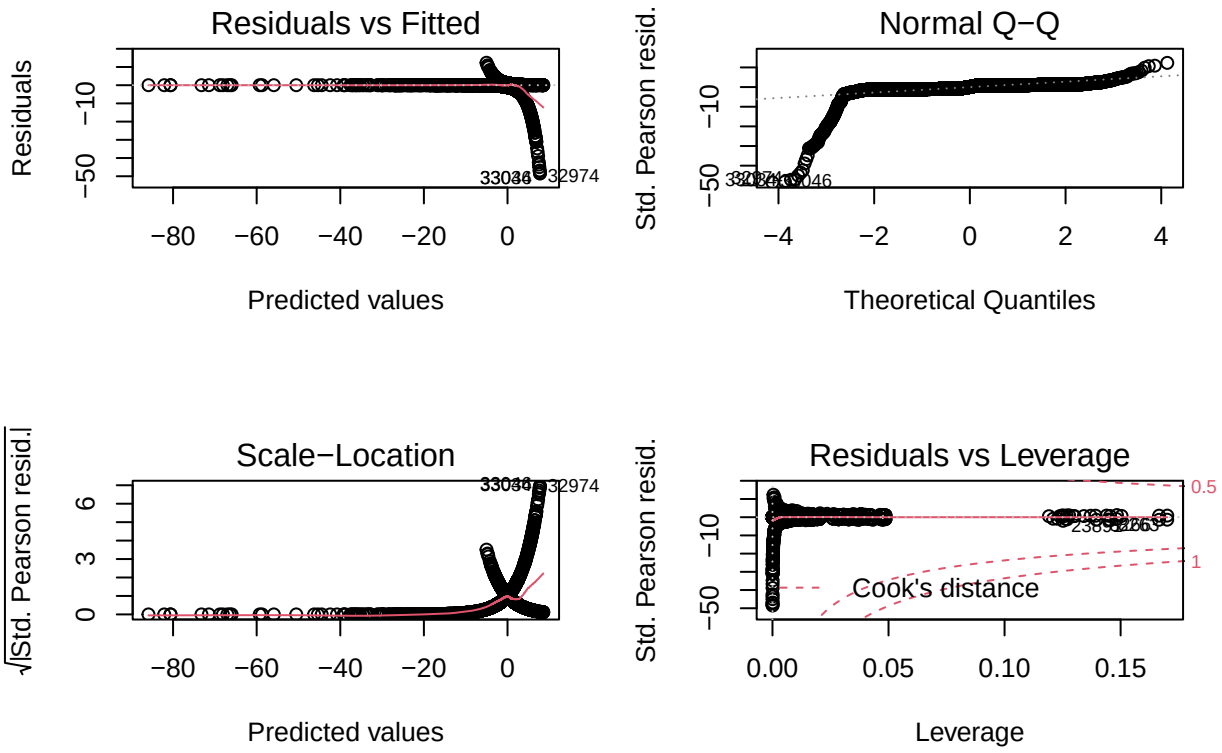
```
## [1] "Male Accuracy:  0.577592900721021"
```

```
options(warn=0)
```

When I factor the female data set, I set female as 1 and male as 0. Using that data set, I split the data into train and test and generate my female Logistic regression model.

```
options(warn=-1)
sdfFemale <- sdf
sdfFemale$Gender <- as.factor(ifelse(sdfFemale$Gender == "female", 1, 0))
split2 <- sample(1:nrow(sdfFemale), nrow(sdfFemale)*0.75, replace=FALSE)
femaletrain <- sdfFemale[split2, ]
femaletest  <- sdfFemale[-split2, ]
```

```
femaleglm <- glm(Gender~.-Year, data = femaletrain, family = "binomial")
par(mfrow=c(2,2))
plot(femaleglm)
```



The results of the graph are basically the same as the male, just mirrored onto the other side.

```
femaleprobs <- predict(femaleglm, newdata = femaletest)
femalepred <- ifelse(femaleprobs > 0.5, 1, 0)
femaleacc <- mean(femalepred == femaletest$Gender)
print(paste("Female Accuracy: ", femaleacc))
```

```
## [1] "Female Accuracy:  0.583028286189684"
```

```
options(warn=0)
```

The reason why I made two models is I wanted to see if there would be a difference in which Gender I used to make the model. The results came back with the female model having a slightly higher accuracy than the male, showing that even though the change was minute, both models did not yield the same results.

##Naive Bayes

I divided the data set into train and test and used it to build the Naive Bayes model.

```
library(e1071)
set.seed(1234)
```

```r
split1 <- sample(1:nrow(sdf), nrow(sdf)*0.75, replace=FALSE)
train <- sdf[split1, ]
test  <- sdf[-split1, ]
nbsdf <- naiveBayes(Gender~., data = train)
nbsdf
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
## female   male
## 0.5022 0.4978
##
## Conditional probabilities:
##         Country
## Y             Albania Antigua and Barbuda    Argentina      Armenia
##   female 0.0073626859        0.0090561037 0.0126638198 0.0107495214
##   male   0.0075020426        0.0083190968 0.0129243111 0.0095818168
##         Country
## Y               Aruba    Australia      Austria   Azerbaijan      Bahamas
##   female 0.0067736710 0.0113385363 0.0119275512 0.0064791636 0.0085407157
##   male   0.0060907673 0.0121815346 0.0135185323 0.0076505979 0.0089133180
##         Country
## Y             Bahrain     Barbados      Belarus      Belgium       Belize
##   female 0.0072154322 0.0104550140 0.0095714917 0.0121484317 0.0100868797
##   male   0.0075763203 0.0096560945 0.0095818168 0.0115873134 0.0102503157
##         Country
## Y             Bermuda Bosnia and Herzegovina       Brazil Brunei Darussalam
##   female 0.0007362686           0.0033868355 0.0128847003      0.0061846562
##   male   0.0005942212           0.0029711060 0.0117358687      0.0061650449
##         Country
## Y            Bulgaria   Cabo Verde       Canada Cayman Islands        Chile
##   female 0.0122956855 0.0003681343 0.0114121632   0.0004417612 0.0113385363
##   male   0.0109930922 0.0004456659 0.0110673698   0.0003713882 0.0121072569
##         Country
## Y            Colombia   Costa Rica      Croatia         Cuba       Cyprus
##   female 0.0112649094 0.0120748049 0.0103077603 0.0076571933 0.0052275070
##   male   0.0107702592 0.0118101463 0.0107702592 0.0084676521 0.0048280472
##         Country
## Y        Czech Republic      Denmark     Dominica      Ecuador        Egypt
##   female   0.0101605066 0.0069945516 0.0022824326 0.0125901929 0.0063319099
##   male     0.0095818168 0.0077248756 0.0022283295 0.0118101463 0.0063878779
##         Country
## Y         El Salvador      Estonia         Fiji      Finland       France
##   female 0.0089088499 0.0109704020 0.0038285967 0.0094242380 0.0121484317
##   male   0.0097303721 0.0112159251 0.0035653272 0.0101017604 0.0119587016
##         Country
## Y        French Guiana      Georgia      Germany       Greece      Grenada
##   female  0.0053011339 0.0107495214 0.0080253276 0.0124429392 0.0086879694
```

```
##   male    0.0048280472 0.0101017604 0.0085419297 0.0126272005 0.0088390403
##         Country
## Y          Guadeloupe    Guatemala       Guyana Hong Kong SAR      Hungary
##   female 0.0045648653 0.0115594169 0.0092033574  0.0117066706 0.0128110735
##   male   0.0052737131 0.0115130357 0.0091361509  0.0121815346 0.0127014781
##         Country
## Y             Iceland Iran (Islamic Rep of)      Ireland       Israel
##   female 0.0117802975          0.0009571492 0.0128847003 0.0122220586
##   male   0.0113644804          0.0009656094 0.0127757558 0.0117358687
##         Country
## Y               Italy      Jamaica        Japan   Kazakhstan     Kiribati
##   female 0.0120011780 0.0072890590 0.0125165660 0.0106022677 0.0036077161
##   male   0.0125529228 0.0067592661 0.0131471440 0.0110673698 0.0039367154
##         Country
## Y              Kuwait   Kyrgyzstan       Latvia    Lithuania   Luxembourg
##   female 0.0104550140 0.0113385363 0.0120011780 0.0103077603 0.0122956855
##   male   0.0101017604 0.0115873134 0.0128500334 0.0117358687 0.0122558122
##         Country
## Y               Macau     Maldives        Malta   Martinique    Mauritius
##   female 0.0002945074 0.0037549698 0.0122956855 0.0058901487 0.0123693123
##   male   0.0003713882 0.0031939389 0.0121072569 0.0053479908 0.0111416475
##         Country
## Y             Mayotte       Mexico     Mongolia   Montenegro  Netherlands
##   female 0.0013989103 0.0129583272 0.0003681343 0.0035340892 0.0122956855
##   male   0.0011884424 0.0124786452 0.0004456659 0.0035653272 0.0122558122
##         Country
## Y         New Zealand    Nicaragua       Norway         Oman       Panama
##   female 0.0116330437 0.0015461640 0.0104550140 0.0009571492 0.0107495214
##   male   0.0113644804 0.0021540518 0.0106217039 0.0009656094 0.0109930922
##         Country
## Y            Paraguay  Philippines       Poland     Portugal  Puerto Rico
##   female 0.0108231483 0.0049329996 0.0101605066 0.0104550140 0.0127374466
##   male   0.0111416475 0.0051251578 0.0101760380 0.0106959816 0.0121815346
##         Country
## Y               Qatar Republic of Korea Republic of Moldova      Reunion
##   female 0.0049329996      0.0100132528        0.0110440289 0.0047857458
##   male   0.0050508802      0.0098789274        0.0109930922 0.0043081037
##         Country
## Y           Rodrigues      Romania Russian Federation Saint Kitts and Nevis
##   female 0.0040494772 0.0094978648       0.0117802975          0.0009571492
##   male   0.0035653272 0.0096560945       0.0106959816          0.0010398871
##         Country
## Y         Saint Lucia Saint Vincent and Grenadines   San Marino
##   female 0.0111912826                 0.0087615962 0.0022824326
##   male   0.0112902028                 0.0095075392 0.0025997177
##         Country
## Y        Sao Tome and Principe       Serbia   Seychelles    Singapore
##   female          0.0010307760 0.0064055367 0.0061110293 0.0123693123
##   male            0.0008913318 0.0060907673 0.0056451014 0.0118101463
##         Country
## Y            Slovakia     Slovenia South Africa        Spain    Sri Lanka
##   female 0.0069945516 0.0111912826 0.0069209248 0.0120011780 0.0051538801
##   male   0.0069820991 0.0108445369 0.0066849885 0.0123300899 0.0057193790
##         Country
```

```
## Y            Suriname       Sweden  Switzerland TFYR Macedonia     Thailand
##   female 0.0114857900 0.0099396260 0.0066264173   0.0070681785 0.0114121632
##   male   0.0103988710 0.0100274827 0.0065364332   0.0077248756 0.0114387581
##         Country
## Y        Trinidad and Tobago       Turkey Turkmenistan      Ukraine
##   female        0.0116330437 0.0024296863 0.0108231483 0.0112649094
##   male          0.0098046498 0.0023026071 0.0109188145 0.0103988710
##         Country
## Y        United Arab Emirates United Kingdom United States of America
##   female         0.0019142983   0.0116330437             0.0118539243
##   male           0.0024511624   0.0125529228             0.0125529228
##         Country
## Y             Uruguay   Uzbekistan Venezuela (Bolivarian Republic of)
##   female 0.0106022677 0.0094978648                         0.0105286408
##   male   0.0112159251 0.0103245933                         0.0104731486
##         Country
## Y        Virgin Islands (USA)
##   female         0.0053011339
##   male           0.0064621555
##
##         Year
## Y            [,1]     [,2]
##   female 1998.867 10.14969
##   male   1998.959 10.17844
##
##         Age_Group
## Y        05-14 years 15-24 years 25-34 years 35-54 years 55-74 years 75+ years
##   female   0.1661022   0.1640406   0.1681637   0.1651450   0.1685319 0.1680165
##   male     0.1669019   0.1671990   0.1638565   0.1671247   0.1689074 0.1660105
##
##         Number_of_Suicides
## Y            [,1]     [,2]
##   female 104.5262  323.878
##   male   341.4408 1167.166
##
##         Population
## Y           [,1]    [,2]
##   female 1737024 3721154
##   male   1662019 3646989
```

```
nbpred <- predict(nbsdf, newdata = test, type = "class")
nbacc <- mean(nbpred == test$Gender)
print(paste("Accuracy: ", nbacc))
```

```
## [1] "Accuracy:  0.526677759290072"
```

The accuracy of this model came out to 53%, which is a little less than that of the logistic regression models.

##SVM:

Lastly, the SVM model. First, I am trying to find the best value for cost. To do so, I split the data into train, test and validation and using those set to tune and find the best cost.

```
library(e1071)
set.seed(1234)
index <- sample(seq(1, 3), size = nrow(sdf), replace = TRUE, prob = c(0.8, 0.1, 0.1))
train <- sdf[index == 1, ]
test <- sdf[index == 2, ]
validation <- sdf[index == 3, ]
stune <- tune(svm, Gender~., data = validation, kernel = "linear", ranges = list(cost = c(0.001, 0.01, 
summary(stune)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##   100
##
## - best performance: 0.4010517
##
## - Detailed performance results:
##    cost     error dispersion
## 1 1e-03 0.4874199 0.03544743
## 2 1e-02 0.4603679 0.03927038
## 3 1e-01 0.4220036 0.03286901
## 4 1e+00 0.4292026 0.02742886
## 5 5e+00 0.4115406 0.02533823
## 6 1e+01 0.4085034 0.02486540
## 7 1e+02 0.4010517 0.02491031
```

For some reason, the best parameter remained empty and I couldnt find the best value for cost. So, I used a random value, 100.

```
ssvm <- svm(Gender~., data = train, kernal = "linear", cost = 100, scale = TRUE)
summary(ssvm)
```

```
##
## Call:
## svm(formula = Gender ~ ., data = train, kernal = "linear", cost = 100,
##     scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  100
##
## Number of Support Vectors:  21085
##
##  ( 10565 10520 )
##
##
```

```
## Number of Classes:  2
##
## Levels:
##  female male
```

```
pred <- predict(ssvm, newdata = test)
table(pred, test$Gender)
```

```
##
## pred     female male
##   female   1569  753
##   male      287  963
```

```
svmacc <- mean(pred == test$Gender)
print(paste("Accuracy: ", svmacc))
```

```
## [1] "Accuracy:  0.708846584546473"
```

Results:

The ranking for the best to worst algorithms for this particular data set is as follows:

1. SVM with cost 100
2. Both Logistic Regression Models
3. Naive Bayes

SVM performed the best for this data set. It had the highest accuracy with 71%. The logistic regression models came second with the female model having an accuracy of 58% and male model with an accuracy of 57%. Lastly, the Naive Bayes model with an accuracy of 53%.

Overall, these three algorithms did an alright job at classifying the data. SVM easily did the best job out of the three. If there is one thing that I would want to look into more, is why the tuning for SVM didn't return a best cost value.