Analysis of Gun Violence in America Using Machine Learning

Group 22: Maximiliano Alvarado, Sara Michael, Luis Waldo

April 28, 2023

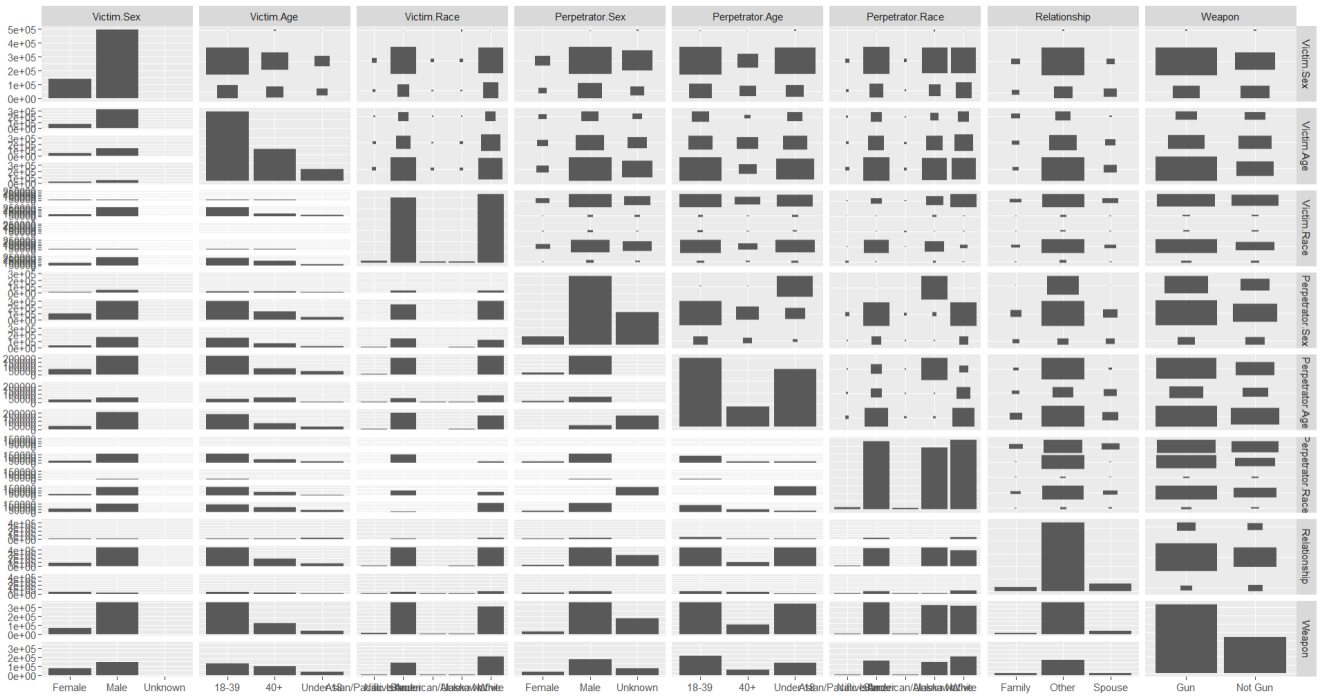
Introduction

Using data we extracted from The Murder Accountability Project, the most complete database of homicides in the United States from 1976 to 2014, we aimed for this report to answer the following question; *Depending on the case variables, can we predict if the victim was killed by a gun?* The case variables we used to predict if it was a gun or not being the Victim and Perpetrator's sex, age, race, and the relationship between them. We then had the weapon variable as our response; "Gun" or "Not Gun". All variables we used were categorical. We are going to use the Logistical Regression Model and the Decision Tree Model.

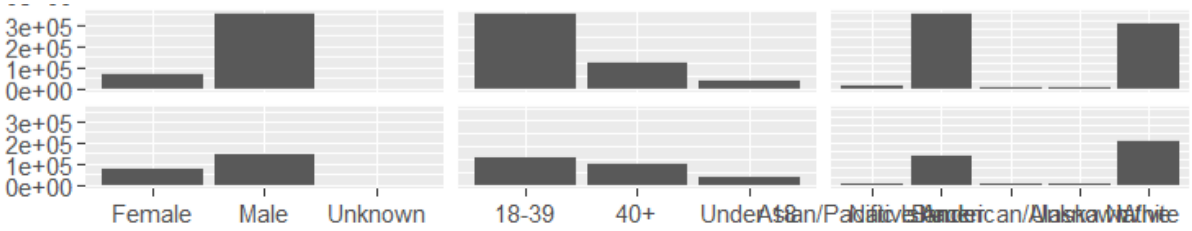
Methods

We first wanted to visualize all the variables compared with each other, then mainly looking at the comparisons between all variables with the Weapon variable. We did this using GGplot2 and GGally.

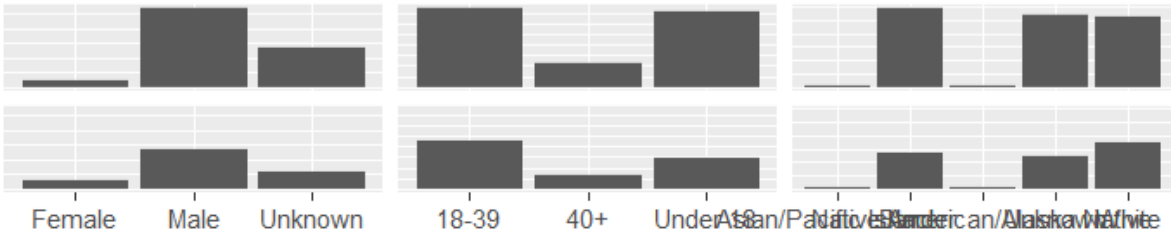
```
library(ggplot2)
library(GGally)
ggpairs(HomicideData)
```



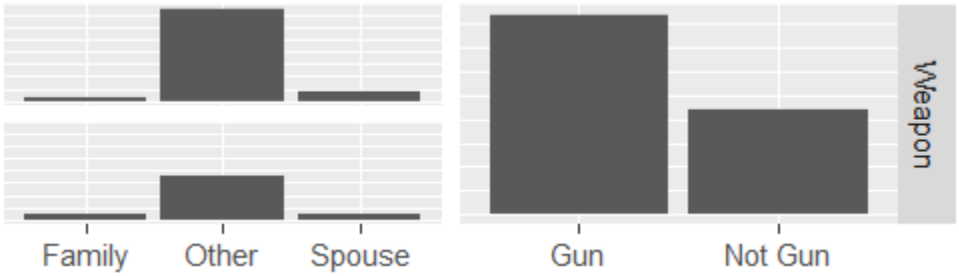
Victim's Sex, Age, and Race compared to *Weapon* (top charts "Gun", bottom "Not Gun")



Perpetrator's Sex, Age, and Race compared to *Weapon* (top charts "Gun", bottom "Not Gun")



Relationship and *Weapon* compared to *Weapon* (top charts "Gun", bottom "Not Gun")



The plots can visually show that almost all variables have some sort of correlation with the variable Weapon, more strongly with the Victim and Perpetrator's Sex and race, and their relationship between each other.

Logistic Regression Model

Starting with the Logistic Regression Model, we chose to do this model because the response variable (*Weapon*) is a categorical variable denoted by its binary nature (Gun/Not Gun). The advantages of this model is that it is easy to implement, makes no assumptions regarding classes within each predictor, provides a measure of predictor importance, and it is less likely to overfit the data. Disadvantages of using logistic regression include assumptions of linearity between independent and dependent variables and sensitivity to outliers. In addition, overfitting and inaccuracy are widely common when this method is used on high dimensional datasets. The formula for the logistic regression model is denoted below by the **glm** function `[glm(response~ ., data = train, family = "binomial"]`.

After cleaning the database, we executed the following code to perform the Logistical Regression Model 10 times using different random seeds, randomly splitting our data into 80% training set and 20% testing set:

```
for(i in 1:10) {
  #Splitting data
  set.seed(i)
  sample = sample.int(n = nrow(HomicideData),
                      size = floor(.8*nrow(HomicideData)),
                      replace = F)
  train = HomicideData[sample,]
  test = HomicideData[-sample,]
  #Performing the logistic regression model with all variables
  Homicide.glm = glm(Weapon ~ .,
                     data = train,
                     family = "binomial")
  print(summary(Homicide.glm))
  glm.pred = predict(Homicide.glm,
                     newdata = test,
                     type="response")
  yhat = ifelse(glm.pred < 0.5, "Gun", "Not Gun")
  print(table(yhat,test$Weapon))
}
```

On the first FOR loop, we acquired the summary and confusion table:

SUMMARY:

```
Call:
glm(formula = Weapon ~ ., family = "binomial", data = train)

Coefficients:
                                          Estimate Std. Error  z value Pr(>|z|)
(Intercept)                               1.556587   0.038393   40.543  < 2e-16 ***
Victim.SexMale                           -1.054944   0.007908 -133.400  < 2e-16 ***
Victim.SexUnknown                        -0.120283   0.081588   -1.474  0.14041
Victim.Age40+                             0.790730   0.007196  109.892  < 2e-16 ***
Victim.AgeUnder 18                        0.704765   0.010952   64.348  < 2e-16 ***
Victim.RaceBlack                         -0.360520   0.028912  -12.469  < 2e-16 ***
Victim.RaceNative American/Alaska Native  0.531793   0.049079   10.836  < 2e-16 ***
Victim.RaceUnknown                        0.178129   0.042411    4.200 2.67e-05 ***
Victim.RaceWhite                          0.121902   0.028653    4.254 2.10e-05 ***
Perpetrator.SexMale                      -0.999317   0.012391  -80.645  < 2e-16 ***
Perpetrator.SexUnknown                   -0.627391   0.036451  -17.212  < 2e-16 ***
Perpetrator.Age40+                       -0.497531   0.010073  -49.391  < 2e-16 ***
Perpetrator.AgeUnder 18                  -0.679572   0.012274  -55.368  < 2e-16 ***
Perpetrator.RaceBlack                     0.064482   0.037263    1.730  0.08355 .
Perpetrator.RaceNative American/Alaska Native  0.589320   0.058506   10.073  < 2e-16 ***
Perpetrator.RaceUnknown                   0.135166   0.049505    2.730  0.00633 **
Perpetrator.RaceWhite                    -0.043960   0.037020   -1.187  0.23505
RelationshipOther                        -0.516449   0.015072  -34.266  < 2e-16 ***
RelationshipSpouse                       -1.061281   0.017819  -59.558  < 2e-16 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 658013  on 510761  degrees of freedom
Residual deviance: 597237  on 510743  degrees of freedom
AIC: 597275

Number of Fisher Scoring iterations: 4
```

CONFUSION MATRIX:

```
yhat        Gun Not Gun
  Gun     76984   32205
  Not Gun  6881   11621
```

We can see that on the first FOR loop, we got an error rate of 0.3061. From all 10 FOR loops, we got an average error rate of 0.30686. This means our Logistical Regression Model was able to correctly predict if the victim was killed by a gun or not about 70% of the time. We didn't choose to leave out any predictors and redo the model because looking at the summary output, each variable had most of their levels with a very small p-value. Although some levels of some predictors had a large p-value, it didn't justify removing the entire predictor.

Decision Tree Model

Our second model is using the Decision Tree Model, we chose to do this model because the dimensions of the dataset we used are rather large and this model is well suited for assessing vast data. Since the response is a binary, categorical variable, we must assert that this is a classification tree when building the model. Advantages of this model are that it is easy to interpret, takes less data preparation and manipulation, uses a non-parametric algorithm which means it needs few (if any) assumptions for classification, and can be used for non-linear problems. Disadvantages of using this model include its tendency to overfitting data, it requires feature reduction since it cannot handle too many features, and using this model can produce high variances and alterations to the data may heavily impact the accuracy of predictions. The formula for the Decision tree model is (include that part here)

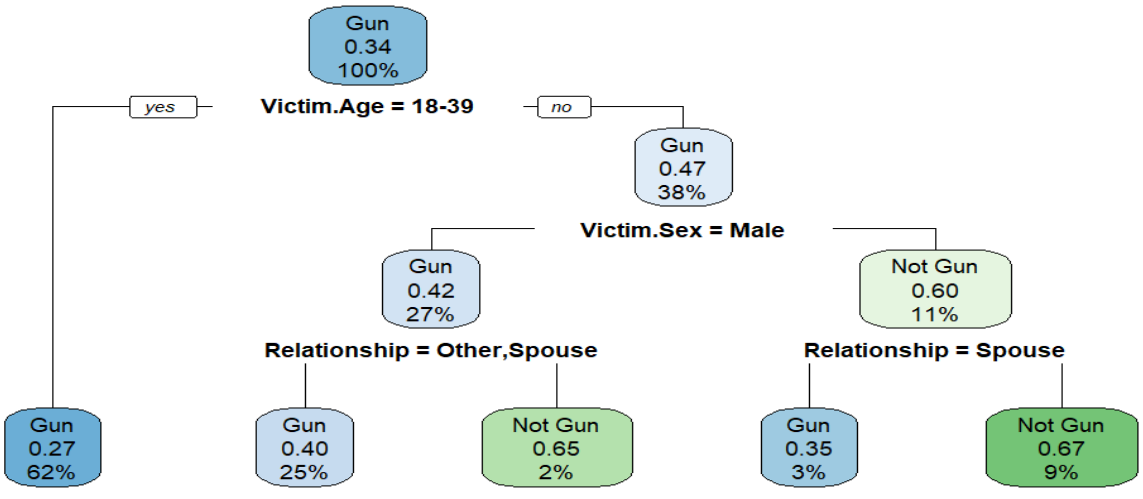
After cleaning the database, we separated the data into training and testing sets. In the code below, the **set.seed** function is used to set the random number generator's seed. In this case, the RNG's seed is set to 1. Then, the **createDataPartition** function was used which observed *Weapon* as the dependent variable. In this, the training set contains 80% of the observations and the testing set contains 20%.

To create the decision tree, we utilized the **rpart** and **rpart.plot** functions within the **rpart** and **rpart.function** packages. The **rpart** function was used by inputting a formula that specifies the relationship between the variables (*Weapon* and the predictors) and a data frame including the training data. Then, the predictor space was recursively partitioned into subsets based on the predictor values using an impurity measure (like the Gini index) to determine the best splits. Each node contains a predictor that yields the greatest impurity reduction and proceeds to create two branches that correspond to potential values of said predictor. This process continues repeatedly until a minimum number of observations in a subset are met or a maximum tree depth is reached. Following this, the **rpart.plot** function was used to create the visual representation of the tree. The completed tree can be used to predict the value of the response variable (*Weapon*).

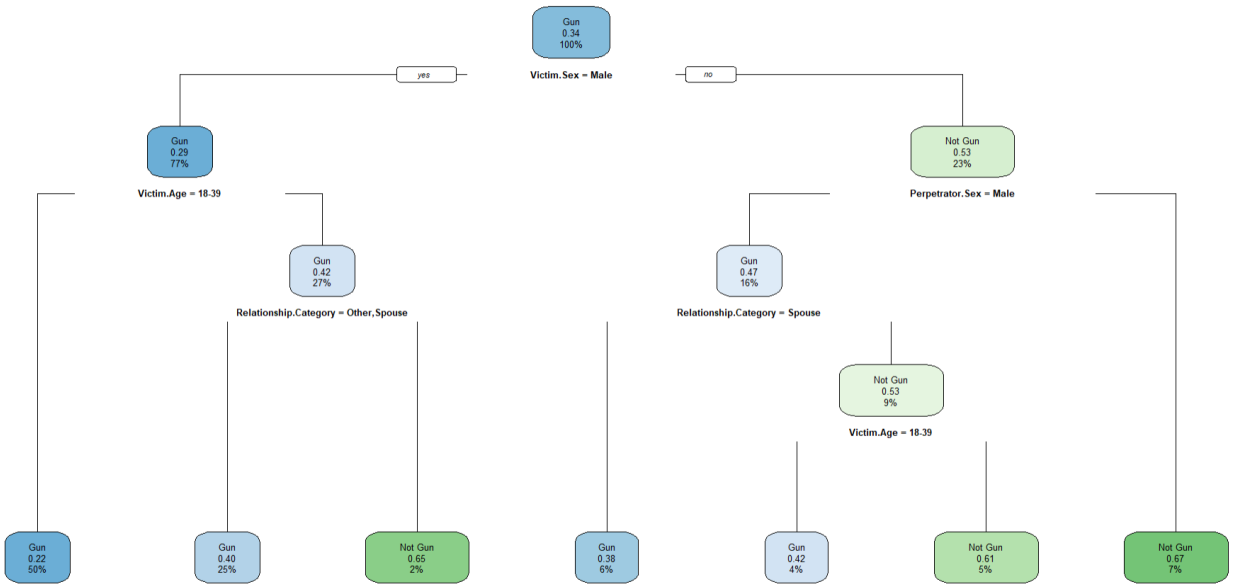
```
for (i in 1:10){
      set.seed(i)
      index <- createDataPartition(HomicideData$Weapon, p = 0.8, list =
      FALSE)
      train <- HomicideData[index,]
      test <- HomicideData[-index,]
      library(rpart)
      library(rpart.plot)
      HomicideData.rpart <- rpart(Weapon ~ ., data = train)
      rpart.plot(HomicideData.rpart)
      pred <- predict(HomicideData.rpart, newdata = test, type =
      "class")
      library(caret)
      confusionMatrix(pred, test$Weapon)
}
```

We then acquired the following decision tree plot, confusion matrix, and statistics: We plotted 10 trees in total (one for each seed()) But they all have the same general structure as the two below.

PLOT: 1



PLOT: 2



CONFUSION MATRIX AND STATISTICS:

```
Confusion Matrix and Statistics
          Reference
Prediction   Gun Not Gun
   Gun     79044   34593
   Not Gun  4662    9391

               Accuracy : 0.6926
                 95% CI : (0.69, 0.6951)
    No Information Rate : 0.6555
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.1882

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9443
            Specificity : 0.2135
         Pos Pred Value : 0.6956
         Neg Pred Value : 0.6683
             Prevalence : 0.6555
         Detection Rate : 0.6190
   Detection Prevalence : 0.8899
      Balanced Accuracy : 0.5789

       'Positive' Class : Gun
```

As seen above, the trees generated make use of: “Victim.Age”, “Victim.Sex”, “Perpetrator.Sex” and “Relationship” which are used to predict the *Weapon* variable. For every tree of the 10 generated, the accuracy is consistently around 70%. The main difference amongst them is what they choose as their first split criterion. Roughly half will choose Victim.Age as the most significant input variable while the other half choose Victim.Sex. Trees that choose one as most significant will include the other as the second split criterion to be considered. We can conclude that the age and sex of the victim are equally significant in determining whether one is the victim of gun violence.

Although the accuracy rate is provided as an output of the confusionMatrix() function, we can calculate it using the matrix provided; it is equal to the accurate values of the matrix divided by the total contents of the matrix. The error rate is the proportion of incorrect predictions made by a model and can be obtained by taking the complement of the accuracy rate (1-accuracy rate). In addition, the sensitivity and specificity rates are 0.9443 and 0.2135, respectively. From the tree, we can conclude that the most significant input variables in determining the response are Victim.Age, Perpetrator.Sex, Victim.Sex, and Relationship. The most significant of these variables would be Victim.Age and Victim.Sex as these two are consistently labeled as the very first split criterion.

Conclusion

With the classification tree, we see that two of the most important factors in predicting gun violence would be the victim’s age and sex. The two groups most at risk of victimization are males as well as people between 18-31 years old. For male victims, we see that they are less likely to be victimized by their spouse as they are by somebody with whom they have no intimate relationship with. While females are not as likely to be victimized, they are significantly more likely to be the victim of homicide by their spouse than by anybody else. However, this is more likely to be done through means other than a firearm. We can extrapolate from this that female homicide victims are largely the result of domestic violence. Males on the other hand make up a large majority of gun violence victimization by the hands of other males.

We have observed a misclassification rate of about 30% that is consistent among both models.

Bibliography

GeeksforGeeks. (2023, April 19). *Decision tree in R programming*. GeeksforGeeks. Retrieved April 28, 2023, from https://www.geeksforgeeks.org/decision-tree-in-r-programming/

GeeksforGeeks. (2023, March 31). *Logistic regression in machine learning*. GeeksforGeeks. Retrieved April 28, 2023, from https://www.geeksforgeeks.org/understanding-logistic-regression/

Project, M. A. (2017, February 10). *Homicide reports, 1980-2014*. Kaggle. Retrieved April 28, 2023, from https://www.kaggle.com/datasets/murderaccountability/homicide-reports