

DS 4420: Machine Learning and Data Mining II
Spring 2021

Project Checkpoint #2

Project Title: Analyzing Mass Shooting Data

Team Members: Ryan Liang



Table of Contents

- I. Introduction
 - A. Problem Description
 - B. Motivation
- II. Dataset
- III. Methods
- IV. Results
 - A. Classification
 - 1. Task #1
 - 2. Task #2
 - 3. Task #3
 - B. Regression
 - 1. Task #4a
 - C. Classification (Revisited)
 - 1. Task #4b

Introduction

- Problem Description

Is it a classification or regression problem?

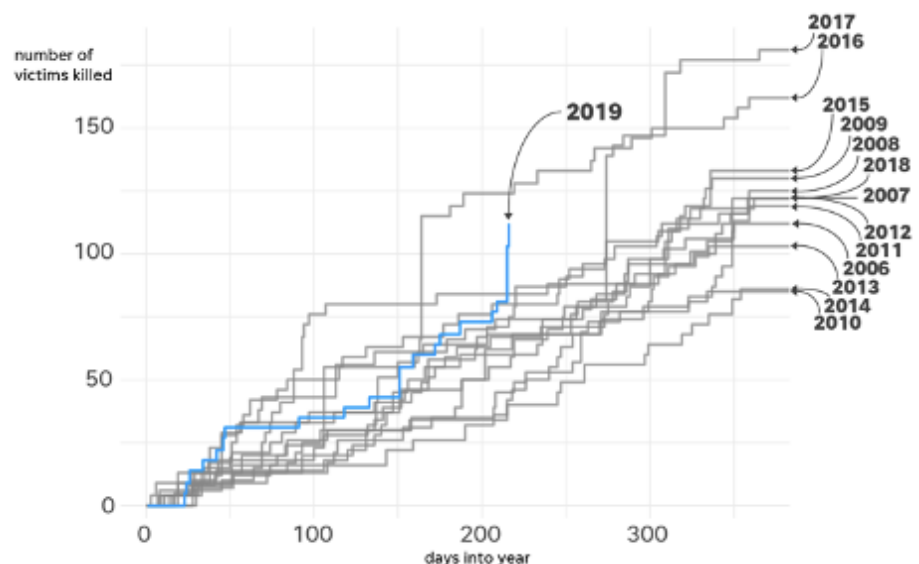
There are two approaches to my problem regarding how machine learning will be conducted. I plan to process the problem as mainly a classification problem with a later attempt at regression.

Classification: My machine learning program will take in the variables that indicate the shooter's race, gender, and mental health status, and classify the data into each category based on these variables.

****Regression:** A regression task I can attempt to handle is predicting the total number of victims killed based on the variables being used (mentioned above)

Why is it important? Describe the motivation

Mass shootings in the U.S. are a serious problem, with 2019 showing record high numbers of both fatalities and shootings in just the first half of the year (as pictured in the graph). As we enter further into 2021, the effort to reduce the number of mass shootings has been aided negligibly, and almost nothing is getting better. The goal of the project will be developing key insights about the victims and also the perpetrators of mass shootings, and analyzing the mental health, race, location, and motivation (and any other important information) about the killers and where the mass shootings take place. When tackling a project of this magnitude, it's important to keep in mind the sensitivity of it, and how important it is to not to draw generalizations and conclusions about race and/or gender, as that is what fuels stereotypes and dangerously leads to racial profiling. However, it is also important to study and analyze key patterns in order to get a full comprehensive picture and to understand the problem better, especially when the issue is one of this caliber, and especially when enough resources haven't been allocated to studying the problem in the first place.



Dataset

There is one major dataset that I'm using. This dataset is the MassShootingsDatasetVer5.csv, and in it, we have information about the shooter, their race, their age, and other general demographic information. We also have several columns including location, description, summary, fatalities, injured, and # of victims total. The columns that we do not have to encode as numerical variables are the columns that already contain numerical values, which in this case are # of victims total, and the age of the victim. However, we encounter a problem when we want to run several machine learning models on the dataset, which happens to contain a lot of string variables such as race. The way I dealt with this problem was by encoding those several variables as numerical values.

For example, the variable "Mental Health Issues" was a column in our dataset that indicated whether the shooter suffered from Mental Health Issues during or before the time of the mass shooting. This was previously encoded as a variable with values "Yes", "No", "Unknown", etc., which is not translated well when running through a machine learning model. So in the form of preprocessing, I created a new list that would contain all the entries, but this time decoded as numerical values, and then iterated through the Mental Health Issues column and if the variable at the current entry was "Yes", I would encode that entry as "1", and for "No" it would be decoded as "0". I did this for the entire column, and I decoded/preprocessed other variables similar to this such as "Race" and "Gender". Mental Health and Gender ended up being binary variables (save for the "Unknown / 2" variable)

A good machine learning project isn't complete unless the data has been cleaned AND preprocessed/normalized properly, and so, in order to measure the performance of the machine learning models, I used the built-in StandardScaler() model to normalize and preprocess the data.

Finally, I am splitting up the project by creating different tasks/problems for me to solve. The way I have split up this project is by the following tasks:

- 1) Predicting the gender of the shooter (**Classification**)
- 2) Predicting the mental health status of the shooter (**Classification**)
- 3) Predicting the race of the shooter (**Classification**)
- 4) Predicting the total # of victims of a mass shooting (**Regression**)

I split the data up this way:

1. **Training/Testing data** (easily changeable)
 - a. This was done using the built-in **train_test_split()** method from sci-kit learn
2. **Features/Labels**
 - a. i.e., Problem #1 is split up as:
 - i. **Features** = Race, Mental Health Issues, Total victims
 - ii. **Label** = Gender

Methods

Describe your implemented methods and preliminary prospects. Given your current progress, what next steps are you planning?

The main methods that I have chosen for this project are running several classification models from scikit-learn and measuring their performance. The models that I have chosen are:

- Naive Bayes
- Logistic regression
- kNN
- Decision Tree
- SVM (Support Vector Machine)

And for Regression:

- Linear Regression

After running these models through the training data features and labels and then testing them on the testing data, analysis is done through several common metrics. So, before jumping into analysis, the metrics I had decided to use are:

- Accuracy
- Error
- Precision
- Recall
- ...which culminates into a Confusion Matrix
- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)

Results

The problem/tasks are split up in the following ways:

- Task/problem
- Problem type
- Models used
- Results
- Analysis
- *Other notes/miscellaneous*

Task/Problem #1: Predicting the *gender* of the shooter

Problem type: *Classification*

Models used:

- Naive Bayes
- Logistic Regression

Results:

<i>Model</i>	<i>Data</i> (% = size)	<i>Accuracy</i>	<i>Error</i>
Naive Bayes	Training (35%)	~96%, (0.9611650485436893)	~4%, (0.03883495145631066)
	Testing (65%)	~97%, (0.9663461538461539)	~3%, (0.033653846153846145)
Logistic Regression	Training & Testing	~98%, (0.979381443298969)	~2%, (0.020618556701030966)

Analysis:

Both classification models were able to predict the gender of the shooter with very high accuracy (typically over ~95%). Additionally, after interaction with the data, it can be seen that relatively high portions of the data show that the shooter's gender tends to be Male.

Initially, other metrics were included to measure the performance of the models, including precision, recall, F1 score, and AUC/ROC scores. However, for a problem like this, there was an **overwhelming amount** of data that ended up being *male* (only one record was a female shooter), and so even when the models assigned a label of *female* to unknown data (on the test set), the false positive rate (FPR)/true positive rate (TPR) would not be a good way to measure the model because there would be only one instance where a TPR or True Positive would not be 0, resulting in both division by 0 errors and a 0 for precision, recall, and F1 score, etc. What I mean by this is that let's say that the one record where the shooter was female was in the held out test data set. If the model(s) predicted that one record in the test data to be *male* rather than *female* (a likely scenario, considering the training data would be considerably male), then the TPR/FPR would end up being 0 because we never predicted the female record accurately.

Task/Problem #2: Predicting the *Mental Health Status* of the shooter

Problem type: *Classification*

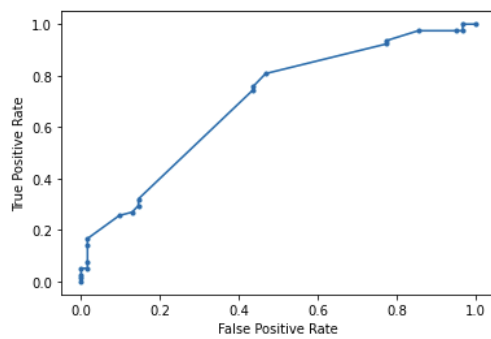
Models used:

- Naive Bayes
- Logistic Regression
- kNN (k-Nearest Neighbors)

Results:

<i>Model</i>	<i>Data</i> (% = size)	<i>Accuracy</i>	<i>Error</i>
Naive Bayes	Training (30%)	~66%, (0.6610169491525424)	~34%, (0.3389830508474576)
	Testing (70%)	~61%, (0.6142857142857143)	~39%, (0.3857142857142857)
Logistic Regression	Training & Testing	~56%, (0.5642857142857143)	~44%, (0.4357142857142857)
kNN ($k = 9$)	Training & Testing	~54%, (0.5428571428571428)	~46%, (0.4571428571428572)

Additional Metrics:

<i>Model</i>	<i>Metric</i>
Logistic Regression	Precision = ~74%, (0.7428571428571429)
	Recall = ~33%, (0.3333333333333333)
	F1 Score = ~46%, (0.4601769911504425)
	AUC = ~70%, (0.6950992555831265) 

Analysis:

The classification models were able to predict the Mental Health Status of the shooter with less accuracy than gender, but not with extremely negligible results. Our models performed with about ~55% - ~65% accuracy. What this tells us is that it is not as easy to determine whether a shooter is suffering from mental health issues (based on the given variables), but it is not a factor to rule out. Based on an initial exploration of the data, more often than not, the shooter is suffering from some type of mental health issue, which likely means that we are predicting with a ~60% - ~65% accuracy that the shooter is suffering from a mental health issue. This is important because this means that mental health care is a sector that we should invest more resources into as a society, as it is determined to be one of the factors that can lead to mass shootings and gun violence.

The kNN model was run through multiple iterations, with different sizes k to ensure that we were using the optimal k size to maximize performance. The accuracy and error of each k value was recorded and reported, and then the maximum accuracy was stored and used. In the Naive Bayes model, the AUC/ROC score was used to measure the FPR and TPR for the model. The AUC is an important metric to plot true and false positive rates and further back up the performance of the model. An AUC score close to 1 is an ideal/optimal rate, and a 0.5 score indicates a random classifier. Our model scored ~0.7, which is a considerably good AUC score. This backs up the decent performance of the Naive Bayes for this specific task. Finally, for Logistic Regression, there was >1 instance of either label/class, and so we were able to report our precision, recall, and F1 score. All of these metrics are reported above under *Results*.

Task/Problem #3: Predicting the *race* of the shooter

Problem type: *Classification*

Models used:

- kNN (k-Nearest Neighbors)
- SVM (Support Vector Machine)

Results:

<i>Model</i>	<i>Data</i>	<i>Accuracy</i>	<i>Error</i>
kNN	Training & Testing	~25%, (0.2532467532467532)	~75%, (0.7467532467532467)
SVM	Training & Testing	~21%, (0.2077922077922078)	~79%, (0.7922077922077921)

Analysis:

Our classification models were **not** able to predict the race of the shooter with very high accuracy. Our models performed with about ~20% - ~25% accuracy, and so what this tells us is that it is not easy to determine the race of the shooter solely based on the variables we are given (mental health status, gender, and total number of victims). Interestingly enough, we actually can eliminate the variable of **gender** because, as previously explored, over ~99% of the data ended up being male shooters, and so this would not allow our race variable to be accurately predicted based off of the gender variable. This leaves us mental health status and total number of victims as our real feature/predictor variables.

In the future, to get a better grasp on this specific task, we should use more variables than just the ones that we encoded. Some variables of interest within the dataset may be *target*, *location*, *employed*, *age*, and *cause*. Additionally variables will give the models we train more features to base our predictions off of and may improve performance.

Task/Problem #4a: *Predicting the **total number of victims** of a mass shooting*

Problem type: *Regression*

Models used:

- Linear Regression

Results:

<i>Data</i> (% = size)	<i>Metric</i>	<i>Value</i>
Training Data (50%)	MSE	2101.6067217792
	RMSE	45.84328436946027
Testing Data (50%)	MSE	175.3739841254949
	RMSE	13.24288428271934

Additional Metrics

<i>Race</i>	<i>Gender</i>	<i>Mental Health Issues</i>
-3.141506	-2.550480	3.861922

Analysis:

The performance of the Linear Regression model is measured through the MSE and RMSE metric. The MSE is the average of all of the residuals (difference between actual and predicted values) squared. In terms of our actual values, the MSE for both of our data sets is not great, but the RMSEs have redeemable scores. The RMSE measures how concentrated the data is around the regression line, and relates to how far the data points vary from the predicted values on the regression line. The RMSE values for both are not awful - considering what the task at hand was. We were trying to predict the total number of victims, and that number can vary a lot. On average, if we only vary about ~13 on our testing data set, that is not something to be ashamed of. However, the model can definitely be fine-tuned in the future to improve performance.

Additionally, from our Linear Regression model, we were able to derive a coefficient table, which tells us the importance/weight of each variable/feature on the regression line. From this coefficient table, we can see that the variable with most weight on the regression model was Mental Health Issues. This is important and backs up a conclusion from before stating that Mental Health is a really important area to invest more research and resources into, because on multiple occasions it has proven to be a factor that plays a role in increased gun violence and mass shootings.

Task/Problem #4b: *Predicting the **total number of victims** of a mass shooting*

Problem type: *Classification*

Models used:

- Logistic Regression

Results:

<i>Model</i>	<i>Data</i>	<i>Accuracy</i>	<i>Error</i>
Logistic Regression	Training & Testing	~78%, (0.7835051546391752)	~22%, (0.21649484536082475)

Analysis:

The regression task that involved predicting the total number of victims of a mass shooting was transformed into a classification task through some further data processing. The current variable of total victims was placed into *buckets*, i.e., if a record had 14 victims, it was placed into a bucket of 10-19 victims. This was then assigned a value (the value being the index

of the list, i.e., 10-19 victims = index 1)¹. Through this process, we were able to encode the victims column as a variable represented by a single value. This was also important because, as previously mentioned, the total number of victims can be really hard to precisely predict, as the model will be considered wrong if you are just one number off. However, it would be nice to consider this case because the model got pretty close but not exactly right, which is why we are using a bucket technique.

After this process, we ran the task as a typical classification problem. Running a Logistic Regression model through this newly processed data, we see a great model performance of ~78%. This is pretty good performance because we are able to predict the right “bucket” of victims based on the few variables that we have provided to us with over 75% accuracy. Keeping this in mind, in the future this can hold importance because we can use other variables like location and “open” or “closed” location to predict in the real world what areas of the world we should be careful of or keep extreme security on.

¹ This is all explained further in the Jupyter Notebook