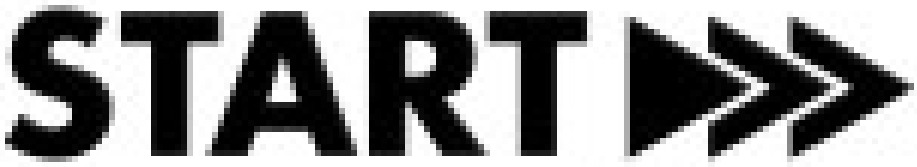


Predicting Attributes of Global Terrorist Attacks Using START Data

Benjamin Cabrera & Andre Perry

Table of Contents

Abstract / Executive Summary

In recent times, terrorism has seen an unfortunate prevalence all over the world. For most nations, there is not a week where one doesn't hear about a terrorist attack whether it's in their country or overseas, whether small or large. Because it's such a relevant topic, we wanted to use machine learning techniques to gain some potentially valuable insights from our analysis that could hopefully contribute to the overall discussion on what can be done.

START is a university-based research organization, and they receive contributions from all over the globe which are used to help with the maintenance of the Global Terrorism Database (GTD). Since 1970 until the present, it has over 200,000 incidents and specific details for each one. There are 135 parameters total, but for our research we were only interested in two. We aimed to accomplish two things: seeing how accurate machine learning methods can predict both whether an attack is successful or not, and whether an attack lasts longer than 24 hours; these ideas are the basis for our research questions. The analysis was done using both Python and R, and we used naive Bayes, classification trees, and logistic regression to create our models. The accuracy would be determined through analyzing confusion matrices.

The dependent variables for each research question were the success of the attack and whether the terrorist incident lasted longer than 24 hours. For the naive Bayes method and classification trees, similar measures were taken to ensure randomization and control. For example, 70-30 cross validation splits were used throughout, and the procedure for each question was generally the same, the only difference being that parameters had to be changed for each question. Similarly, the logistic and penalized regression models used an 80-20 split and 10-fold split for each research question to ensure control. Creating testing and training sets were used for both questions, with the training sets being essential since we must train our model using that set to see how accurate our test set will be. We had to encode categorical parameters into binary outputs for our models, and although the classification tree limited parameters to only the most essential ones, all processes led us to obtain models where we could test the accuracy using a confusion matrix. Our models for the first research question, predicting successful or failed terrorist attacks, were much better than chance alone with results of 74.2% and 84% for the tree method and naive bayes method, respectively. The best model from the logistic regression was found using a logistic regression model between success and property, with an AUC score of 0.70 and 84.25% accuracy. The naive bayes model for the 2nd research question had an

astounding 95.8% accuracy, while the tree method only had a 54% accuracy, just slightly better than chance alone. The best logistic model found for the second research question was a logistic regression model between ishostkid and extended variables, which yielded an AUC score of 0.97 and 96.85% accuracy. Low scores on our AUC curves suggests that our dataset fails to classify observations in the minority classes for both research questions.

Project Plan

Profile of the Organization and Background of the Opportunity

Primary Organization Details:

National Consortium for the Study of Terrorism and Responses to Terrorism (START)

Founded: 2005

Founder: Dr. Gary LaFree

Current Head: William Braniff

Location: University of Maryland, College Park.

Mailing Address:

START, University of Maryland

PO Box Number 266

5245 Greenbelt Rd

College Park, MD 20740

Communication:

Phone: (301) 405-6600

Fax: (301) 314-1980

E-mail: infostart@umd.edu

Organization Description:

The National Consortium for the Study of Terrorism and Responses to Terrorism, or START, is a government-funded, university-based research and education organization that consists of researchers from across the world who study the causes and consequences of terrorism. The organization explores important questions and conducts scientific research using its open-source data, much of which was gathered through unclassified media articles. START has a large focus on education; the organization has developed programs for secondary, undergraduate, and graduate students, and disseminates its findings to homeland security professionals and the public through research, education and training efforts.

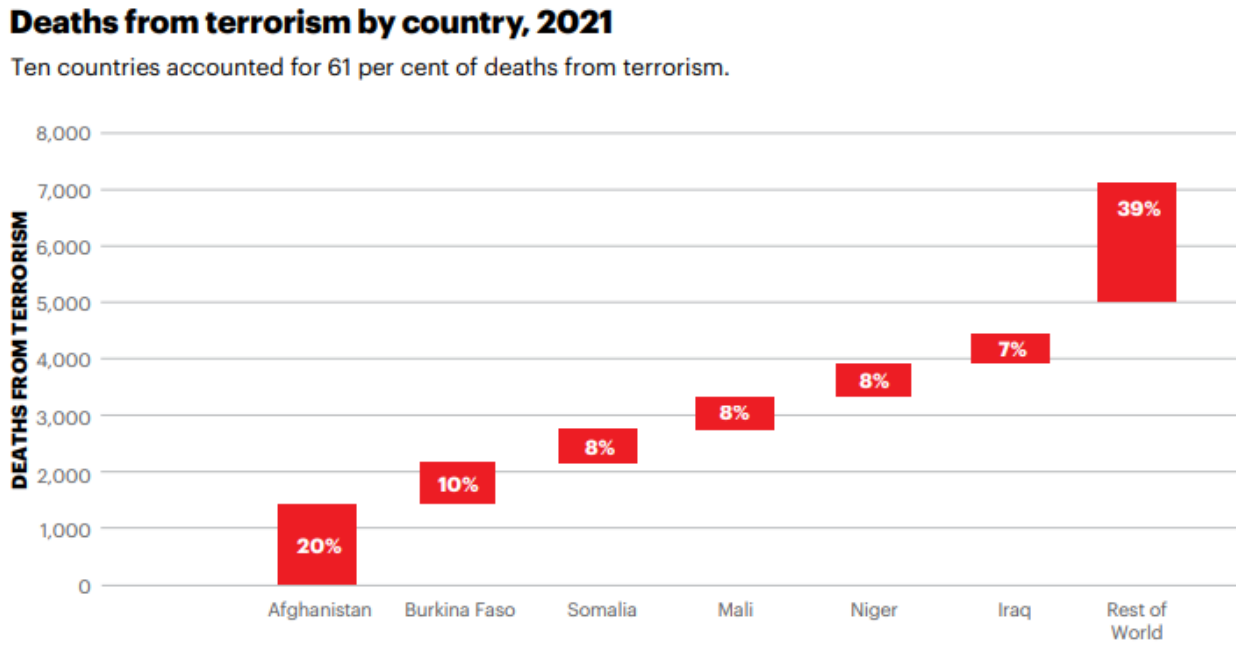
START was established in 2005 under the US Department of Homeland Security Center of Excellence. The goal of the organization is to use contemporary methods to improve the

understanding of the "origins, dynamics, and social and psychological impacts of terrorism." Initially, the organization received a $12 million dollar grant, which was renewed in 2008 in order to continue its outstanding research. Additionally, START was recognized by the DHS for not only its contributions to the United States, but for its efforts in the Global Terrorism Database, the dataset that this project focuses on. START is partnered with more than 75 university, private industry, and national laboratory partners, and while its main source of funding is from DHS, there are numerous other institutions that contribute to its funds. Additionally, international partners like the German Federal Foreign Office and the United Kingdom Foreign, Commonwealth, and Development Office have contributed funding to the organization.

START has clear goals listed as a mission and a vision. Its mission is, "to advance science-based knowledge about the human causes and consequences of terrorism and serve as a leading resource for homeland security policymakers and practitioners"; its vision is to "provide homeland security policy-makers and practitioners with the highest quality, data-driven research findings on the human causes and consequences of terrorism in an effort to ensure that homeland security policies and operations reflect these understandings about human behaviors."

Organization Analysis / Opportunity:

By using the Global Terrorism Database provided by START which contains over 200,000 terrorist attack observations and 135 variables, we will analyze research questions regarding terrorist attacks that have occurred after 1997. We will answer a few research questions by creating Logistic Regression and shrinkage models to predict a binary variable, AUC-ROC curves to assess accuracy, Naive Bayes methods for classification, and also decision trees. These models and assessments will provide insight on the predictive power of these variables in response to the research questions and will hopefully be helpful in understanding the nuances of terrorist attacks. The findings will also benefit national security organizations globally towards the characteristics of terrorist attacks.



Retrieved from https://www.visionofhumanity.org/wp-content/, 2022

Research Questions

Seeing as terrorist attacks can occur anywhere and have detrimental effects on our society and economy, it is important for governments to understand the characteristics of terrorist attacks and their potential effects. The following questions will be investigated in this research project.

RQ1. How accurately can we predict if a terrorist strike succeded or not based on certain parameters?

The aim of this research question is to find out what aspects contribute most to a terrorist strike, successful or not. This information could be useful to governmental figures in order to understand what types of terrorist strikes are most successful which can then be applied towards policy. For example, if a country is shown to have high success rates of assassinations, governmental figures would know to increase security measures. The main parameters we're interested in are the success value of the attack, the South Asia region, the explosives weapon type, the number of people that were slain, whether the incident lasted longer than 24 hours, the number of people wounded, and the bombing/explosion method of attack. We're using the naive bayes method, classification trees, logistic regression and shrinkage models to make predictions.

RQ2. How accurately can we predict whether an attack lasted longer than 24 hours or not?

The intention is to use these key parameters for prediction: the South Asia region, the bombing/explosion method of attack, explosives weapon type, ransom, number of perpetrators, and target type. The main interest is to see if these parameters significantly affect the length of time of the attack. Ransom may play a large role since there could be hostages involved, and the larger the number of people involved, the longer the incident can last. Something we wish to discover is just how significant using explosives affects the length of time. It could be just momentary, or it could have lasting consequences that can last days. We'll use the naive bayes method, classification trees, logistic regression and shrinkage classification techniques.

Hypotheses

H1: A diverse subset from the majority of the database variables will be most relevant to predict the success of a terrorist attack.

Considering that there are many attributes which describe a terrorist attack, there will be feature subsets containing information from a majority of the database variable groups. By exploring what features or categories have the greatest effect in predicting the success of a terrorist attack, we can further split variable subsets to provide better findings.

H2: The inclusion of ransom will increase the likelihood of an attack lasting longer than a day.

The number of people involved and the general method of attack may be decent factors at answering this question, but the key variable will be ransom. With ransom there are likely hostages involved, so the incident may be longer than normal.

Data

The data comes from the START's Global Terrorism Database which was collected through publicly available media articles and electronic news archives, as well as through existing data sets and secondary source materials such as books, journals, and legal documents. The GTD contains 209,707 recorded terrorist incidents which occurred from 1970 to 2020 globally, excluding 1993. The terrorist incident data from the year 1993 was lost prior to the compilation of the GTD. The dataset's variables are all structured and stored in a pre-defined

format. The observations for this dataset includes variables relating to nine categories, which are: GTD ID and Date, Incident Information, Incident Location, Attack Information, Weapon Information, Target / Victim Information, Perpetrator Information, Casualties and Consequences, and Additional Information and Sources.

For the first research question, the main variables of interest are success value of the attack, the weapon type used, the number of people that were slain, whether the incident lasted longer than 24 hours, the number of perpetrators involved, the number of people wounded, and the general method of attack. It contains a mix of both numerical and categorical variables, with the output of interest being the latter. The two categorical variables with more than a binary output are the method of attack and the weapon type. For attacks, the methods are assassination, hostage taking (kidnapping), hostage taking (barricade incident), bombing/explosion, facility/infrastructure attack, armed assault, unarmed assault, hijacking, or unknown. For the weapon type, it includes explosives, incendiary, firearms, chemical, melee, sabotage equipment, vehicle, fake weapons, radiological, biological, or other. These specific input variables were chosen because they seem like the most likely predictors for what constitutes a successful attack or not.

For the second question, the output variable is whether an attack lasted 24 hours or not, with the parameters being region, the general method of attack, weapon type, ransom, number of perpetrators, and what was attacked. Ransom is a binary variable. All of these parameters may have an effect on the length of the attack; they were deemed appropriate to use for this question.

Measurements

For this research project, each of our research questions will look into the specific measurements of success, attack length, and damage occurred. Fortunately, the GTD dataset provides variables for each of the measurements that we wish to investigate. For success, the team at START have defined successful terrorist attacks in this dataset according to it having tangible effects. Therefore, success is not based on the larger goal of the terrorist perpetrator. For example, a terrorist group that kidnaps civilians will be considered successful as long as they assume control of a civilian. This does not necessarily mean that the terrorist's ideological goal has succeeded, however. For each type of terrorist attack, there is a specific criteria to determining success. The length of the attack which we hope to investigate is any attack lasting

more than 24 hours. There is a binary variable *extended* which we will use as our measurement for classification. Finally, the damage occurred for a terrorist attack will be measured through weighing variables in the Casualties and Consequences category.

Methodology

The first research question is attempting to see how accurately our predictions of a successful terrorist attack are. Because there are only two results, either "yes" or "no," this means our method must be chosen with a categorical output in mind. This is why Logistic Regression seems most appropriate since it is the categorical analog to linear regression with numerical variables. However, the Naive Bayes classification and classification tree fulfills the same objective since it is also a method that predicts the value of a categorical output based on numerous inputs, so we will compare the results of these methods. These methods will also be used for the second research question.

Using L1 and L2 regularization, we can shrink the number of variables in our classification. This will allow us to remove any irrelevant variables and better understand our research questions. If this shrinkage improves upon our Logistic Regression, we will apply it towards our research. Tuning our hyperparameters for these regularizations is also something that we will apply in hopes of finding the ideal variable subset.

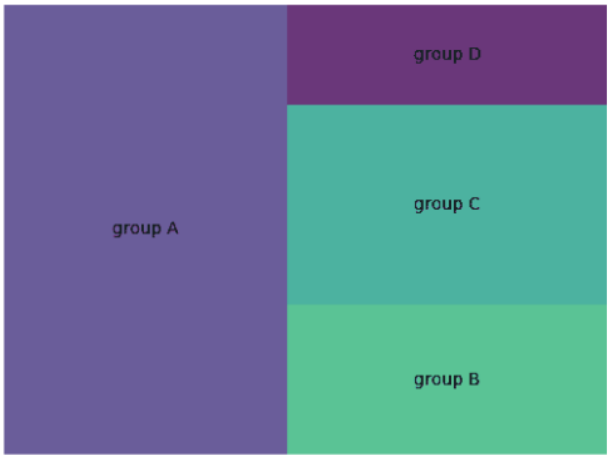
Computational Methods and Outputs

For the questions concerning a categorical output, in order to illustrate the competency of our prediction, we will include a classification metric for the Naive Bayes classifier known as the confusion matrix. The confusion matrix provides us with the true positive, false positive, true negative, and false negative values that our predictions yielded. From there, we can calculate and measure the accuracy and precision of the results. For Logistic Regression and L1 and L2 regularization, we will use the receiver operator characteristics (ROC) curve in tandem with the area under the ROC curve (AUC) score, and accuracy in order to measure the performance. The output for the first question is the predictions for a successful or unsuccessful terrorist attack, and the second question's output will be the predictions for an attack lasting longer than 24 hours. 1 will correspond to a "yes", while 0 will correspond to "no."

Output Summaries

RQ1. How accurately can we predict if a terrorist strike succeded or not based on certain parameters?

The analysis will show how accurately we can predict successful or unsuccessful terrorist attacks. We will illustrate our findings by including a table of confusion matrix values for each classification method, Naive Bayes, Logistic Regression, and L1 and L2 shrinkage. In addition to this, we will visualize the ideal variable subset using a treemap.



RQ2. How accurately can we predict whether an attack lasted longer than 24 hours or not?

The analysis of this research question will be identifying the variables which lead to the highest accuracy and prediction of the *extended* binary variable. In addition, by analyzing the specific variables which were most relevant towards predicting the *extended* variable per each region of the world, we can determine what variables are specific to each region. A treemap will identify the most relevant variables per region. Finally, a table will help provide confusion matrix values for each of the classification methods used (Naive Bayes, Logistic Regression, and L1 and L2 shrinkage).

Literature Review

As terrorist attacks persist globally, it is important to further our understanding on the impact and success of these attacks by utilizing classification techniques. This project relies on classification modeling in order to answer our research objectives. In this section we further investigate existing research in the field of terrorist behavior and terrorist attack modeling.

Peter Krause (2018), author of "When Terrorism Works" mentions crucial information regarding the definition of a successful attack. First, terrorism is effective when the group meets their intended goals. This can be through inspiring fear, overthrowing those in power, and so forth. He then highlights three categories that can establish an attack to be successful or effective; it is through the organizational, strategical, or tactical objectives that it meets. Respectively, they are defined as having to be organized and not self-interested, having firm control such that the government and citizens do not interfere with the group's motives, and having the ability to carry out an attack with precision; this includes following through with the intended location, amounts of violence, and securing the intended target. One does not need to satisfy each category in order for an attack to be successful. This greatly increases the complexity because not only is there significant variation in what's classified as successful or effective, but there is not an even distribution concerning objective types.

Furthermore, one objective's success can rely on the other. For example, having a successful or effective strategic objective most likely is tied to organizational success. With organizational success comes power in numbers; even people who believe that terrorism is ineffective say that one of the most pivotal elements of a strategic success requires large numbers of members or perpetrators. Most importantly, Krause mentions the Global Terrorism Dataset and raises an important question: are attacks that fail to kill anyone tactically ineffective? About 90% of attacks classified as successes in the dataset are tactical ones, but how does one obtain the knowledge necessary to answer the question since most attacks failed to have slain people according to the GTD? It may be beyond the scope of the data or a serious challenge to attempt at this answer because a thorough analysis would have to be done on each group that launches the attack. However, one fact can be duly noted according to Kraus: tactical success is more likely when the wielders can effectively utilize their weapon.

There are two similar features we are using that are in line with Krause's work: number of members or perpetrators, and the weapon of choice. It's difficult to determine whether one is

skilled with their weapon based on data alone, but at least knowing the connection is beneficial. He does bring up the importance of knowing each individual group, and it is something we must take into account when analyzing and interpreting the results since access to all motives behind groups, especially those classified as unknown, is nearly impossible for this project.

Putting Terrorism in Context : Lessons from the Global Terrorism Database by LaFree et al. is book that interprets statistics from the Global Terrorist Database and puts into context "black swan" events, which are unpredicitible but have had a long lasting impact. The data exploration in this book includes exhaustive graphs plotting variables from the GTD over time, in addition to tables of some variable values. Particularly useful charts that relate to our research objectives can be found in Chapter 7, which investigates the deadliness of attacks. For example, the chapter highlights that a majority of terrorist attacks from the GTD resulted in no deaths - "nearly 54 percent of all terrorist attacks from 1970 to 2012 caused no fatalities". Another key piece of information relating to our research questions is that "more than 70 percent of the terrorist organizations named in the GTD perpetrated all of their attacks within one year." Although this source does not use any classification or predictive modeling in its methodology, the focus on interpretation on terrorist attack statistics means that our research questions are relevant towards advancing the understanding of terrorist activity.

Current research applying machine learning to terrorism includes the work by Toure and Gangopadhyay, which used terrorist incident data collected in real time in order to develop a risk model to predict future terrorist attacks. The methodology for the risk model in this project relies on defined logic in addition to certain rules in order to predict future terrorist incidents. The results from this project show that their prediction model had high recall and precision. In our project, we hope to use the same metrics of precision and recall in order to determine our classification model's success.

Exploratory Data Analysis

In this project we used the Global Terrorism Database from the website https://www.start.umd.edu/gtd/ which consists entirely of structured data. We gathered the data from a csv file in Kaggle. It includes documented attacks from 1970-2017 with over 180,000 observations and 135 variables. First, the variables were reduced to a target subset, a list of variables that we were either using or potentially using. From here, the observations and data varied as it was being cleaned because there were many NaN or other observations that were unevenly distributed among variables. This is why for most relationships examined they will not have equal counts, but the variables might be important for our methodology which is why we will keep them. The following variables were explored to better understand our research:

attacktype1 - A categorical variable with values 1-9 describing the general method of attack used in the incident. The categories of attack types are as follows: 1 = Assassination, 2 = Armed Assault, 3 = Bombing / Explosion, 4 = Hijacking, 5 = Hostage Taking (Barricade Incident), 6 = Hostage Taking (Kidnapping), 7 = Facility / Infrastructure Attack, 8 = Unarmed Assault, and 9 = Unknown

extended - A categorical variable describing whether or not the duration of an incident lasted more than 24 hours, where 1 means that the incident extended more than 24 hours and 0 means that the incident extended less than 24 hours

nkill - Number of confirmed casualties from an incident, which takes into account the deaths of all victims and attackers

nperps = Total number of terrorists participating in the incident

nwound - Number of confirmed non-fatal injuries to both attackers and victims

propvalue = Numerical variable on the value of property damage

region = Categorical variable describing ranging from 1-12 describing the region where the incident occurred. The categories for the variable are as follows: 1 = North America, 2 = Central America & Caribbean, 3 = South America, 4 = East Asia, 5 = Southeast Asia, 6 = South Asia, 7 = Central Asia, 8 = Western Europe, 9 = Eastern Europe, 10 = Middle East & North Africa, 11 = Sub-Saharan Africa, 12 = Australasia & Oceania

success - A categorical variable describing whether or not an attack succeeded according to the tangible effects of the attack. Success is not measured in the larger goal of the incident, therefore an attack like a kidnapping would be counted successful even if it did not result in the larger goal

of the terrorist. 1 means that the attack was successful, while 0 means that the attack was unsuccessful

targtype1 = Categorical variable ranging from 1-22 describing the general type of target/victim in the incident. The categories of the variable are as follows: 1 = Business, 2 = Governmental (General), 3 = Police, 4 = Military, 5 = Abortion Related, 6 = Airports & Aircraft, 7 = Government (Diplomatic), 8 = Educational Institution, 9 = Food or Water Supply, 10 = Journalists & Media, 11 = Maritime, 12 = NGO, 13 = Other, 14 = Private Citizens & Property, 15 = Religious Figures / Institutions, 16 = Telecommunication, 17 = Terrorists / Non-State Militias, 18 = Tourists, 19 = Transportation, 20 = Unknown, 21 = Utilities, 22 = Violent Political Parties

weaptype1 = Categorical variable ranging from 1-13 describing the general type of weapon used in the incident. The categories of the variable are as follows: 1 = Biological, 2 = Chemical, 3 = Radiological, 4 = Nuclear, 5 = Firearms, 6 = Explosives, 7 = Fake Weapons, 8 = Incendiary, 9 = Melee, 10 = Vehicle, 11 = Sabotage Equipment, 12 = Other, 13 = Unknown

The first graph in Appendix A gives a general overview of how the amount of terrorist attacks vary per year. The amount fluctuates up and down from 1970 to about 2005, but after that year the attacks dramatically increase with a large peak of over 17,000 attacks in one year alone. The main purpose was to highlight this dramatic increase because with so many attacks concentrated within such a short period of time, there may be specific reasons for this concentration when compared to previous years. It's something we must keep in mind as we develop our model for our research questions because the results in that area may be highly influential more so than the other years.

The graph in Appendix B explores the relationship between the weapon type and the number of attacks performed. The former parameter is an important part of our research questions, so we wanted to analyze how it's related to terrorist attacks in general, not just successful ones. Explosives and firearms vastly outnumbered the other weapon types in terms of counts. Because one of the goals of terrorism is to inspire fear, these two weapons, alongside being more accessible than the other options such as using chemical, radiological, or biological weapons, are what we expected to see the most. Incendiary and melee weapons are the next significant types, but they are magnitudes less than the top 2 types. About 15,000 attacks were performed with unknown weapons; media coverage and getting access to every detail on an attack is difficult, so this is why there is such a large number that aren't known.

Appendix C shows the number of successful terrorist attacks per region. The bulk of successful attacks occurred in the Middle East & North Africa and South Asia. Aside from the attack year, the attack region could be an important grouping method to consider for our methodology in order to fine tune our classification models.

One of the most important parameters is the method of attack that was used. To illustrate how it was distributed, we created a pie chart to examine what were the most and least common methods used. Bombing, explosions, and armed assault were the most common, and it was previously seen that explosives and firearms were major weapon types, so this lines up with the graph in Appendix B. Armed assault, hostage taking, infrastructure attack, and unknown were about evenly distributed with them making about 11-14% of the attack types. Assassination, unarmed assault, hijacking, and hostage taking (barricade incident) made up either 3% or less of all the terrorist attacks; these are operations that are incredibly risky that would probably result in the death of the perpetrator. It leads us to believe that it's why we see little of these types of attacks.

Starting with the graph in Appendix E, we start to investigate how one of our output variables, a successful terrorist attack, is related to other variables. We were interested in the target type, or what the terrorist targeted whether it be people, places, or things, and how it related to successful attacks. There's a vast amount of business, government, police, military, and most importantly private citizens and property that were subject to a successful terrorist attack. These make up the majority of successful attacks, all ranging in the tens of thousands, while the rest of the target types are only in the thousands or hundreds. So far, we recognize that there are attacks mainly done with explosives and firearms, and that the successful attacks occur in buildings or property and not on people; even though we haven't examined the relationships between each one of these variables, we can draw connections from what we see and have an idea of what the general picture is.

In Appendix F, we created a histogram to analyze the distribution of successful or failed terrorist attacks based on year. We used the density measure to see how frequent these were occuring. For both distributions, they were largely skewed to the left and were generally the same shape with peaks in the last 5 years for both. The frequency peak for successes was about 10%, while the peak for the failures was 3%. There was not a single year since 1970 where the frequency of the failures was greater than the successes.

The graphs in appendices G and I are both boxplots with the same parameter corresponding to our first and second question: the number of perpetrators. The graph for Appendix G examines distribution of perpetrators based on whether the attack succeeded or failed, while the graph for Appendix I focused on the distribution of perpetrators based on whether the attack was longer than 24 hours or not. Both graphs exclude outliers because one attack had an extreme number of perpetrators with a count of 25,000. For all distributions, they were skewed to the right. For successes, there was a range of 0 to 16 perpetrators with a median of 3.5. The failed attacks showed a range of 0 to 6 perpetrators with a median of 2. Successes seem to be associated with greater numbers of perpetrators, while failures include less people. The medians were only separated by 1 and a half perpetrators. For the graph on attack length, one that was less than 24 hours had a median of 3 people with a range from 0 to 13 people. The attacks longer than 24 hours had a range of 0 to 41 people with a median of 6. The spread between the boxplots in this graph is much greater than in the graph based on outcomes. It appears that there is some relationship between a longer-lasting attack and a greater number of perpetrators and with ephemeral attacks having vastly less perpetrators.

The double bar graph in Appendix H shows how ransom and the amount of attacks greater or less than 24 hours are related. Ransom was rarely involved, but it appears that there were more cases of attacks that were longer than 24 hours that had the inclusion of ransom rather than not. There were 68,000 attacks that were less than 24 hours that didn't include ransom, whereas there were 5,975 attacks that were greater than 24 hours that did not include ransom.

Appendices J through M show different counts based on terrorist attack type. Appendix J shows that there were more successful attacks than unsuccessful attacks. We can see that the majority of successful attacks came from armed assaults and bombings/explosions. It can also be seen that hostage taking (kidnapping) incidents and facility/ infrastructure attacks were more likely to be successful than unsuccessful. Meanwhile, Appendix K shows that most attacks involving some sort of property value were facility/infrastructure attacks. It is also important to note that unknown attacks composed a small portion of the total number of attacks involving property damage. As well, the incidents involving the greatest property damage were bombings/explosions. Moving on, Appendix L shows that unknown attacks account for the highest count of kills per attack type, followed by bombings/explosions and then armed assaults. The attack type responsible for the highest number of deaths was a bombing/explosion. Finally,

Appendix M shows us that bombings/explosions resulted in the highest count of wounded people, followed by unknown attacks. Again, it could be observed that bombing/explosion incidents were responsible for the highest number of wounded people. This exploration on attack types helps us draw conclusions on what attack types are most frequent and how they result in success, property damage, death, and injury.

This data exploration helps us draw conclusions on what attributes that successful attacks can consist of. This being greater number of perpetrators and armed assault or bombing/explosion attacks. Meanwhile, for the length of these incidents, we can see that variables like higher number of perpetrators and ransom involvements are more likely to be seen in longer attacks. When developing our classification models, these variables will be useful in order to interpret our findings to real life incident attributes.

Methodology

Q1: Terrorist Incident Classification

We seek to accurately classify the success of a terrorist incident in order to apply this model towards future incidents so that authorities can quickly look for the most relevant features found in successful attacks. We are classifying the outcome of a terrorist incident attack using multiple classification models and interpreting the feature set with highest scores in our metrics. In the process of preparing our data, we first considered that every observation prior to the year 1997 did not collect data on around 20 variables. Therefore, we removed those observations prior to 1997 and were ultimately left with around 95,000 observations to use in our classification. We also make sure to remove any string variables since they are either comments on specific observations or text spellings of specific variable categories, therefore unnecessary. Next we looked through the Global Terrorist Database codebook to accurately encode our categorical features. Once we encoded our variables, we decided to indicate 10-fold cross-validation to avoid overfitting our data. 10-fold cross validation allows for us to split our dataset into 10 folds so that for each fold, we can build our model on 9 out of the 10 folds. Afterwards, we will test the 1 remaining fold for model effectiveness, and then repeat the process so that each fold can be the test set. By averaging the score of our folds, we will get the desired metric for our model. This metric will be the AUC-ROC curve for our Logistic Regression, Ridge, and LASSO models, and the confusion matrix for the Decision Tree and Naive Bayes models. We will then compare the values to find the best prediction.

Modeling techniques:

- Logistic Regression
    - Uses a sigmoid function in addition to a gradient descent algorithm in order to calculate the parameters for a model. Since success is a binary variable, we will be able to predict the output of our test dataset and afterwards look at the coefficients for our features to find the most relevant features, along with plotting the AUC-ROC curve.
- Ridge
    - Ridge regression uses a shrinkage estimator in order to shrink values. This is done by utilizing an L2 penalty which equals the square of the magnitude of coefficients. This technique will produce an output similar to our logistic

regression where we can see the most relevant coefficients and plot the AUC-ROC curve.

- LASSO
    - Similar to Ridge, LASSO uses an L1 penalization in order to shrink values. The difference between LASSO and Ridge is that LASSO can completely shrink coefficient values to 0 through the L1 penalty which is equal to the absolute value of the magnitude of the coefficients. We will rely on the default tuning parameter of 1 to control the strength of our penalty.
- Decision Tree
    - Decision trees categorize observations based on how a set of previous decisions were made. This classification model might be useful due to the large number of categorical variables in our dataset. Therefore, we can see how specific values of the success variable will be split according to other variables.
- Naive Bayes Classifier
    - The Naive Bayes Classifier assumes that the presence of one variable is independent of another variable in order to apply Bayes Theorem to find the posterior probability. This model is useful for our dataset with a large number of categorical variables.

Q2: Extended Incident Classification

We want to know what variables are most influential in classifying an extended incident, which is a terrorist incident lasting longer than 24 hours. Being able to understand the most influential aspects of a long terrorist incident could be useful for authorities engaged in a terrorist incident who will be able to apply our feature set towards the potential length of an attack. We will be applying methods identical to Q1 for this question. The only difference is that we are instead predicting the extended variable since it describes whether or not an attack occurred longer than 24 hours.

Data Visualizations and Analysis

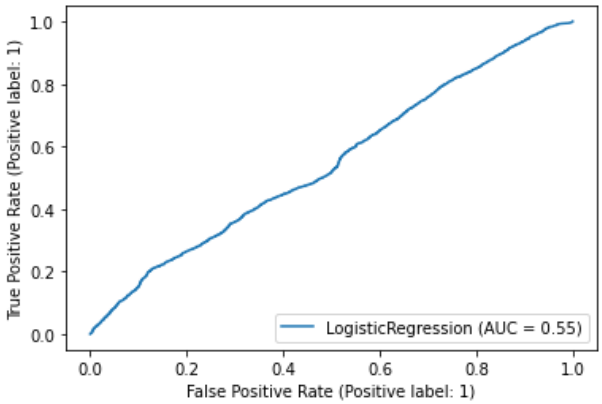
The CSV file for the GTD was requested from the official website for the most up to date version. We performed the same methods for both research questions. For the naive Bayes method, we used training and testing sets with a 70-30 split respectively, then we created a training model in order to test the accuracy of our predictions with the test sets; this is also called cross validation. 70% of the observations were placed into the training dataset while the remaining 30% of observations would be in the testing dataset. These same 70-30 splits were used for classification trees. In order to ensure the results were as authentic as possible, we had to randomize the rows for both the training and testing sets. Meanwhile for our Logistic and penalized regression, we utilized an 80-20 test train split and proceeded by creating a training model with 10 k-fold splits in order to similarly cross validate our models. These 10 k-fold splits allowed for the model to run iterations where the data is split into 10 groups and each iteration, one group gets chosen to be the test dataset while the other 9 remain training datasets. The average of the results after 10 iterations is our final output. After running all of our classification models, confusion matrices were produced. For the logistic and penalized regression models, AUC-ROC curves were also produced.

The independent variables for each question were the success and extended variables, respectively. The success variable is defined as the success of a terrorist incident. Meanwhile the extended variable is defined as whether or not the terrorist incident lasted longer than 24 hours or not.

Question 1

Logistic Regression

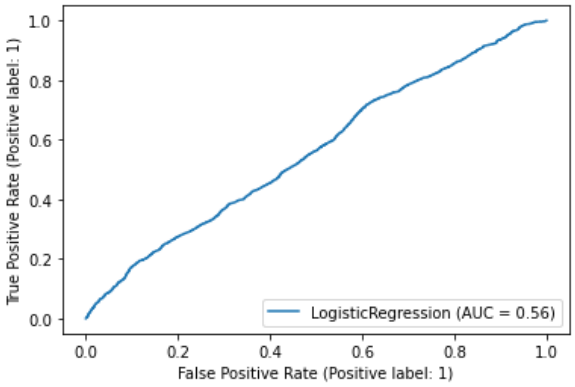
The first step towards answering our research question was to perform a logistic regression using 10-fold cross validation. The intent of the first research is to predict the success of a terrorist incident, and to see what features are most important towards predicting the success. The image below shows the AUC-ROC curve for the model prediction.



The AUC-ROC curve provides a disappointing result. Since the AUC value of 0.55 is so close to 0.5, this means that our logistic regression model cannot effectively differentiate between positive and negative cases for successful terrorist incidents. Therefore, the model does not provide any meaningful information. Looking at Appendix N, we have the confusion matrix which further shows the inaccuracy of the model. An overwhelming number of values (16643) were classified as unsuccessful when in reality they were successful attacks. There were 3258 successfully classified observations, which overall means our model was 16.36% accurate. Looking at Appendix O, the most relevant features towards predicting success were property which is defined by whether or not property damage was involved, ishostkid which is defined by whether or not victims were taken hostage, and nkill which is defined by the total number of fatalities during the terrorist incident. Although the model is not very useful, we have some idea of what aspects are most important in predicting the success of terrorist incidents.

Ridge Regression

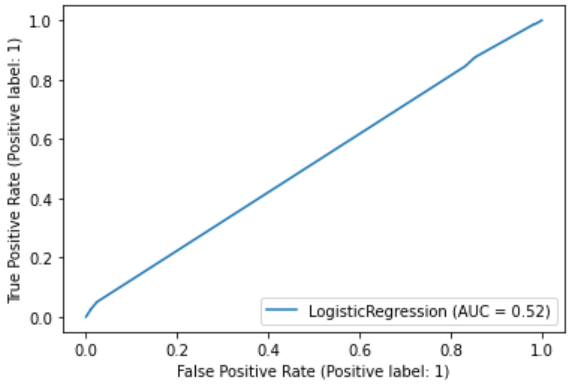
In order to further explore regression, we implemented an L2 penalty in our logistic regression. The results of this Ridge regression were only slightly better than our logistic regression. As you can see below in the AUC-ROC curve, our classification still does not distinguish between successful and unsuccessful terrorist incidents. The confusion matrix in Appendix P shows us that again we are only around 16.339% accurate according to our model. Similar to the logistic regression, we can see in Appendix Q that the top three features towards predicting success were property, nkill, and ishostkid. This model outperformed the other two logistic and penalized models, although not by much.



LASSO Regression

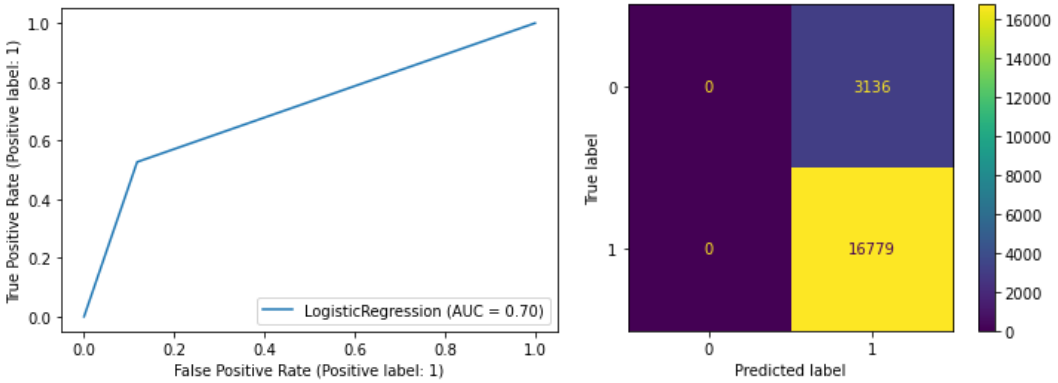
The next attempt to try and improve our logistic regression results was to use an l1 penalty to try and make some variables approach towards 0 so that we can see the main features in predicting success. From the AUC-ROC curve of the LASSO regression, we can see that our results are worse than the two prior regressions but still ineffective in differentiating between successful and unsuccessful attacks. The confusion matrix in Appendix R shows us that the accuracy of this model is 17.95% which is an improvement from the other two models. Meanwhile Appendix S shows us that the only variable with a positive impact in predicting success was nkill. All other variables either approached 0 or were negatively impactful. Overall, these logistic regression models have a difficult time with negative success values which results in low accuracy. Looking at the results between the three models, the only variable present in each classification was nkill. Therefore an impactful variable in classifying successful terrorist incidents is the number of kills taking place during the incident. This means that during a terrorist incident, authorities should hurry to try and end an incident once it involves death. The

variables of property damage and whether or not kidnapping occurred are also important considerations since they were present in two of the three models. Terrorist perpetrators could succeed in their goals if they leave an impact like property damage. They also might receive the attention they desire after holding someone hostage.



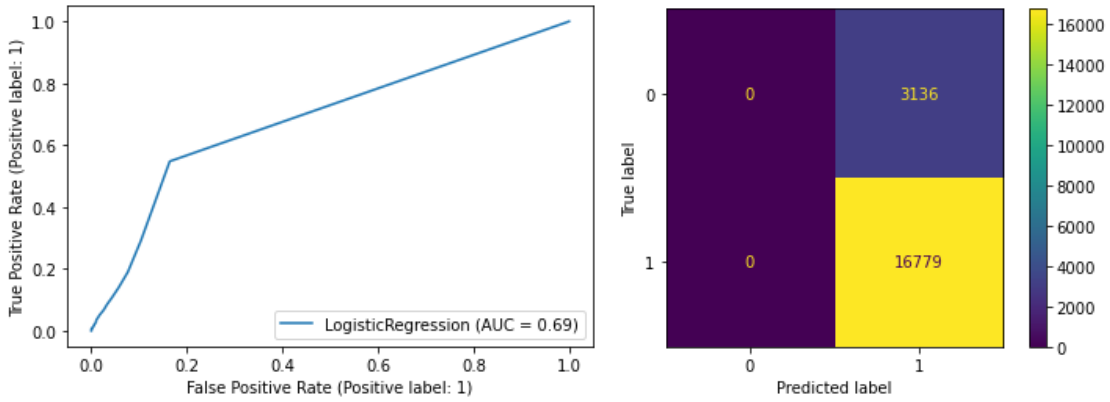
Second Logistic Regression Classification

In an attempt to further improve our results, we performed a logistic regression for each of the top three predictors from our three logistic and penalty models with our target variable. Our first model between success and the property variable yielded the following AUC-ROC curve and confusion matrix:

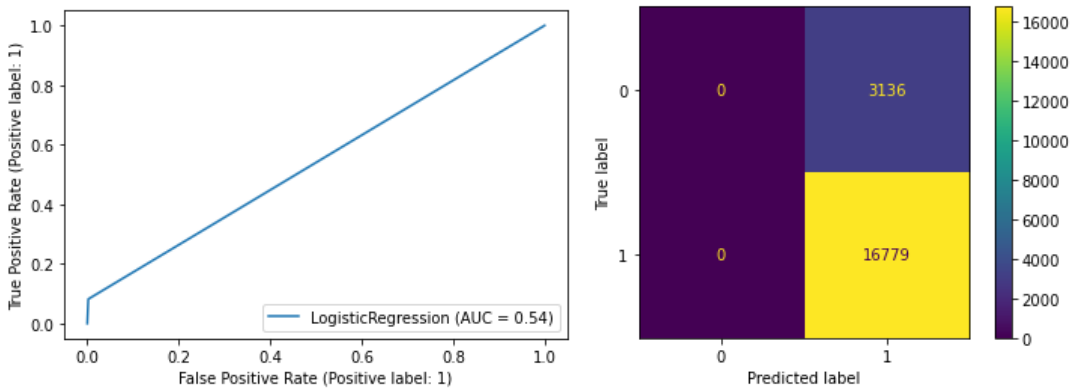


As you can see, this logistic regression led to an AUC value of .7 which is considerably better than our prior model attempts. The accuracy of this model was 84.25%.

Our second logistic regression was between the success target variable and the nkill variable, which resulted in the following AUC curve and confusion matrix:



The results of this logistic regression is similar to the prior model between property and success, with this model being slightly worse due to the AUC score. Finally, we conducted a logistic regression between the target variable success and the ishostkid variable, which led to the following AUC curve and confusion matrix:



This model resulted in an AUC score of .54, which shows that this model had difficulty differentiating between successful and unsuccessful incidents. Overall, this simple insight into the three most relevant variables shows that when using only one variable to predict success, whether or not property damage was involved and the number of kills were useful predictors.

Naive Bayes and Classification Tree

For both research questions, the following table shows the results of using both the naive Bayes and classification tree to measure the accuracy of our model.

Method	Success Accuracy	Extended Accuracy
Full Tree	0.741974	0.540756
Naive Bayes	0.840548	0.957651

We used the parameters number slain, number wounded, the South Asia region, the bombing/explosion attack method, the explosives weapon type, and the extended parameter (whether an attack was longer than 24 hours) to predict whether an attack is successful or not for naive bayes and classification methods. The results show an 84% accurate result for naive bayes, and a 74.2% accurate result for the classification tree. With a difference of about 10%, it's recommended to use the naive bayes method. Classification trees limit the variables used; it helps with bias-variance tradeoff so that our model is neither too basic nor too complex. It limited it to the following: number wounded, number slain, extended, and bombing/explosion. It had a misclassification error rate of 16% out of about 70,000 observations. Although there's less parameters used for the classification tree method, the naive bayes method doesn't include so many parameters that the model is too complex.

The following are the confusion matrices for both methods, respectively.

success	0	1
row_0		
0	0	18
1	4745	25108

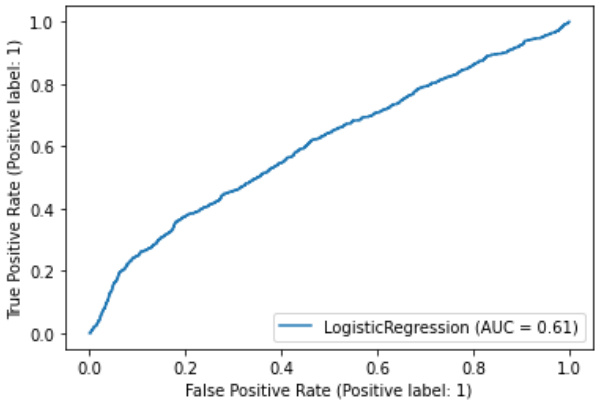
	ytest	
ypred	No	Yes
No	3466	6479
Yes	1229	18699

The accurate predictions are those in which the No (0) and Yes (1) columns and rows match. We can see why the naive bayes method fared better in terms of accuracy; it accurately predicted more observations than the tree method. The tree method was better at predicting a failed terrorist attacks, but it had a much higher chance for error overall. The naive bayes method didn't accurately predict a failed terrorist attack at all, but was very good at predicting the success.

Question 2

Logistic Regression

Similar to Question 1, in Question 2 we want to effectively classify terrorist incidents that occurred more than 24 hours by predicting the 'extended' variable. We used logistic regression to begin, which gave us the following AUC-ROC curve:

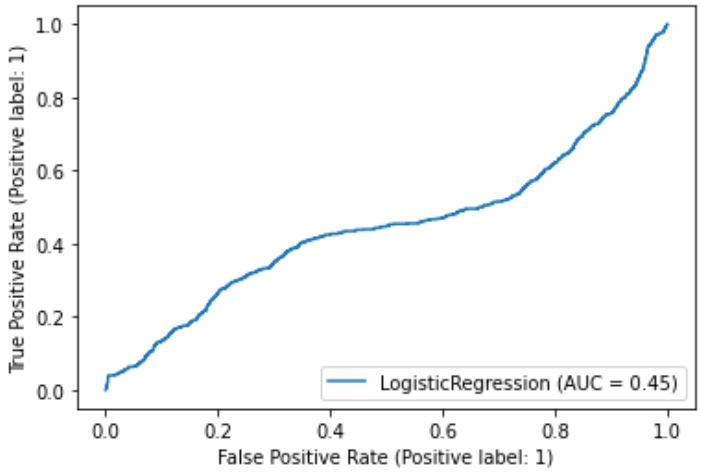


These values are improvements over the prior logistic regression, however the model has a lot of difficulty in successfully determining events that lasted longer than 24 hours. The confusion matrix is somewhat promising since so many true positive values were classified, which means that the accuracy of the model is 95.88%. However the difficulty in predicting true negative values led to the disappointing results in the AUC-ROC curve. The features most relevant towards this prediction include ishostkid, property, and claimed, which is defined by whether or not someone claimed responsibility for the terrorist attack. This logistic regression outperformed the two penalized regression models.

Ridge Regression

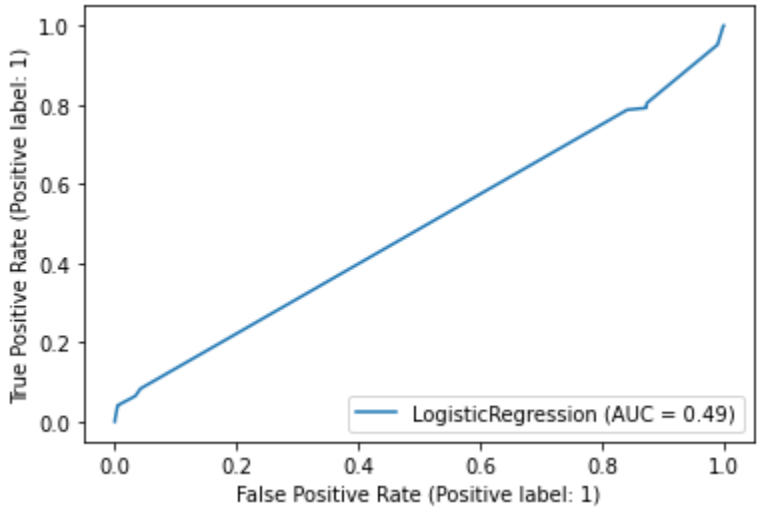
Using an l2 penalty term, we can see in our results that the model is instead predicting the negative extended values as positive and vice versa, which explains why the AUC-ROC curve looks different. The confusion matrix shows us that false positive and true negative values were not predicted at all in the model, which makes the accuracy extremely high at 95.96%. However the same occurrence as the logistic regression is occurring where no negative values were accurately classified, therefore hindering the model. The features most relevant towards this prediction were ishostkid, Hostage Taking (Kidnapping) which is defined by whether or not the

terrorist incident is of the attack type of a kidnapping, and success which is whether or not the attack was successful in its motive.



LASSO Regression

The attempt at LASSO regression for this question yielded the following AUC curve:

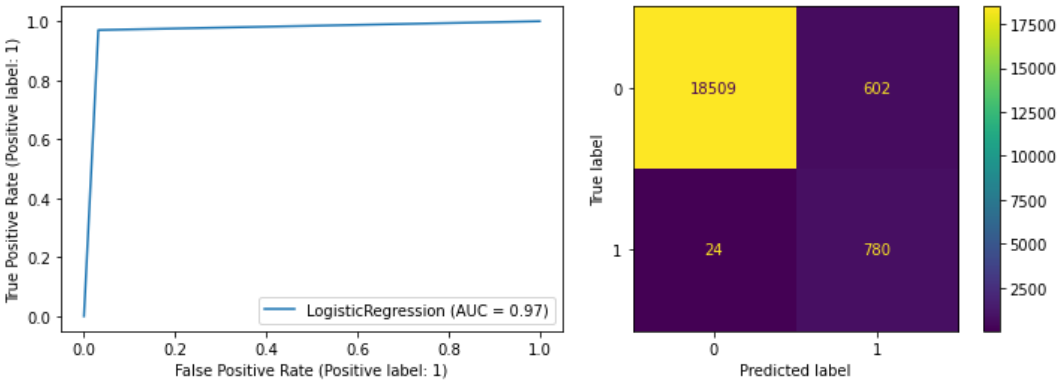


As you can see, the AUC curve is very close to 0.50, which makes the model ineffective in determining whether or not attacks lasted longer than 24 hours. The confusion matrix for this LASSO regression in Appendix T shows that a few negative observations were accurately classified, unlike the other two regression models. The overall accuracy of the model is 95.70%. Meanwhile the only feature which positively impacted the regression was ishostkid, meaning all other features either approached 0 or negatively contributed to the regression. To sum up the three logistic regression models, there is difficulty in predicting true negative cases which might stem from the dataset not having many negative values to predict. Therefore, the models are

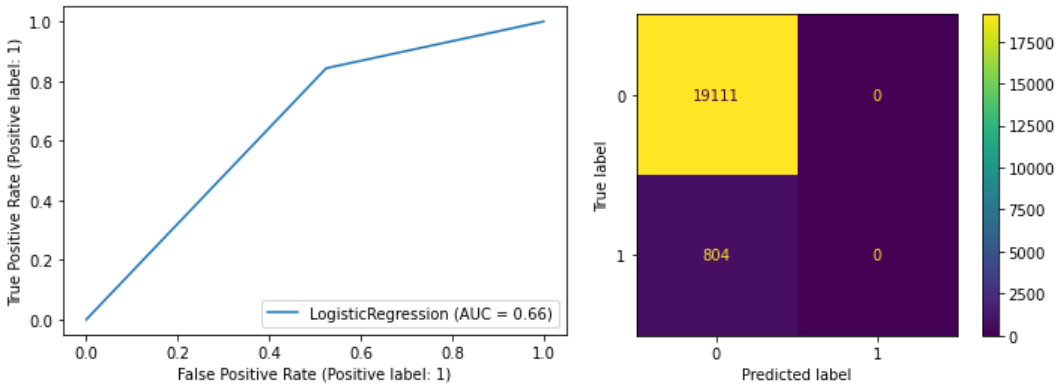
missing the few minority class observations, resulting in mediocre AUC-ROC curves. For all three logistic and penalized models, the ishostkid variable was present in each of the top three variables for the models. This shows us that incidents where terrorists take others hostages is extremely relevant in predicting whether or not an incident will last more than 24 hours. Another notable predictor in this series of models is the claimed variable. Terrorists claiming responsibility could realistically only occur after the incident ends, which means that this variable does not provide much use for future classification during terrorist incidents.

Second Logistic Regression Classification

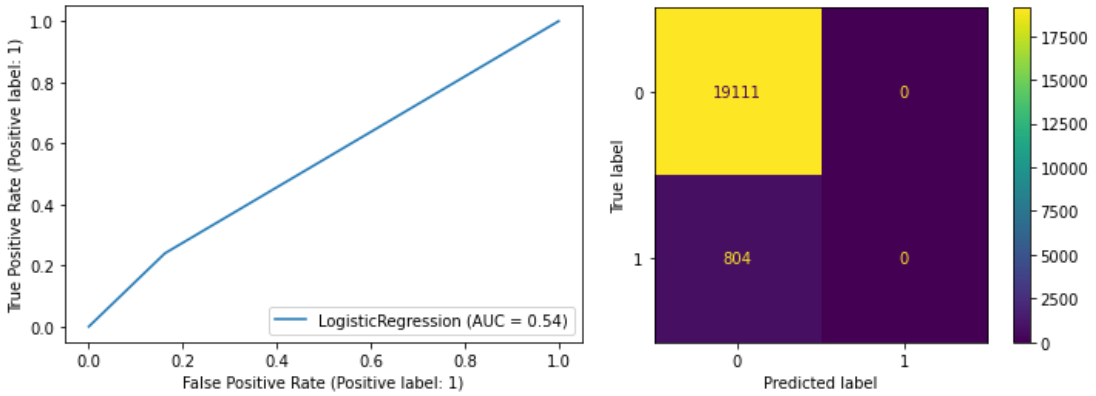
Similar to Question 1, we performed a logistic regression for each of the top three predictors from our three logistic and penalty models with our target variable extended. This is in hopes of improving our prediction by honing in on our most relevant variables. Our first model between the extended and ishostkid variables yielded the following AUC-ROC curve and confusion matrix:



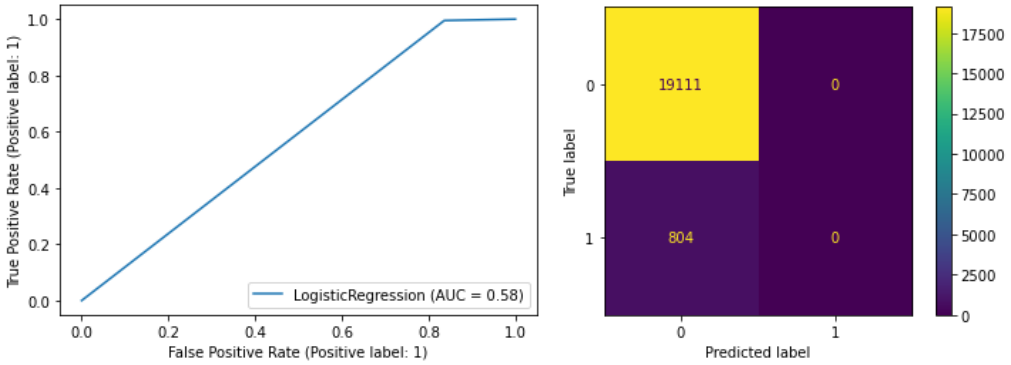
Our second model between property and extended resulted in the following AUC-ROC curve and confusion matrix:



Our third model between ishostkid and extended resulted in the following AUC-ROC curve and confusion matrix:



Finally, our fourth model between success and extended resulted in the following AUC-ROC curve and confusion matrix:



From these results, you can clearly see that the single logistic model between ishostkid and extended was most effective in producing a high AUC score of .97, and a high accuracy of 96.85%. The logistic regression between property and extended produced meaningful results too, since the AUC score of .66 is higher than the prior logistic, LASSO, and Ridge regression attempts. This simple regression shows us that the ishostkid and property variables are effective when used by themselves to predict the target variable of extended terrorist incidents.

Naive Bayes and Decision Tree

This question is focused on the variable “extended”. The parameters used were the bombing/explosion method of attack, target type, South Asia region, and explosives weapon type. Target type includes 20 individual parameters, so the total amount used for the naive bayes

method was 23. This method had an accuracy of 95.8%, while the classification tree had a result of 54%. The confusion matrices are as follows:

extended	0	1
row_0		
0	28549	1144
1	121	57

	ytest	
ypred	No	Yes
No	14939	14
Yes	13705	1215

The classification tree limited the variables from 23 to 3 -- bombing/explosion, military target type, and police target type. The naive bayes method included all the parameters, so it's a highly complex model. This means that it's fitted very well for our training data, but when we use this as a model to predict future attacks, it may be a lot less accurate. The classification tree has few parameters used, so it's a more biased model, meaning the variance will not be as high when predicting data outside what we used. From a pure accuracy standpoint, it is better to use the naive bayes method, but from a pragmatic standpoint we must consider what exactly someone is looking for: are they concerned with all types of targets to see what will make an attack last longer, or do they only want to look at a small subset of them? If the latter is the answer, these two methods will have to be performed with only those few target types they are concerned with.

Ethical Recommendations

Our research on terrorist incidents globally provides ethical implications towards the understanding on what aspects are most important in predicting successful and extended terrorist incidents. Certain groups could be vulnerable to profiling if their features are labeled as important towards predicting the likelihood of successful and extended terrorist incidents. The results of our classification modeling could impact attitudes towards certain groups if our interpretations are not explicitly defined and elaborated upon.

The data usage in this project consists of attributes including terrorist target types, victim type, and victim/perpetrator nationality. Through the data cleaning process of our project, we removed some of these variables due to missing values for a large amount of observations. Regardless, it would be important to screen for these type of variables in order to avoid any discriminatory interpretations from our modeling. If we were to find significance based on nationality from our classification models, this could put certain groups at risk. They could be targeted and deemed terrorists, which is irrational thinking.

While we did use the parameter “region” in order to see if the South Asia location had any significant effects, we adamantly chose not to include the parameter “nationality.” This would cause a multitude of problems, such as othering, reinforcing certain stereotypes, and having the implication that certain people are inherently predisposed to performing terrorist attacks. It’s arguable whether or not using “region” can contribute to othering or certain stigma; we were interested in learning just about a single region, but perhaps using longitude and latitude would have been better in this regard.

If one were to use our results, it should be noted that they must not let implicit bias affect their perception; they must only look at the data that’s presented and make valued judgements based only on that and not on other preconceived notions. We do not condone attributing terrorism with a classification of people or certain parts of the world as part of their identity where it is seen commonly.

The results from our modeling have an important impact in how we interpret the success and extended length of terrorist incidents. Being able to see which variables were most impactful towards predicting our two target variables in our research questions helps us understand the most important attributes of successful and extended terrorist incidents. Therefore, it is important that we highlight only the most impactful variables between all of our classification models.

Governmental officials could see how impactful our model's variables are in predicting successful and extended terrorist incidents and potentially help populations vulnerable to terrorist attacks by implementing policies. This type of research applying classification modeling into terrorist incidents is new in this field, therefore our results should be backed up with further investigation before being used.

The potential impact from not using this research could be a lack of understanding towards how to spot potentially successful or extended terrorist incidents. By not utilizing the results from our research, officials might not know what to look out for when trying to end terrorist incidents.

Challenges

A notable challenge from this project was our data cleaning process which required a large number of observations and variables to be removed. It was important for us to have a dataset without missing values or unencoded variables so that our classification models could function properly. While cleaning our dataset, over half of our variables were removed due to being repeated text formats of numerical variables, or due to the variables not being recorded for a majority of the observations. For example, the country variable had over 200 categories, so instead we relied on the region variable which grouped up countries into regions since it had less categories.

Determining what type of success -- organizational, tactical, or strategical -- that was mentioned in the literature review section was incredibly difficult to define when actually working with the data. While it was stated that most successes were tactical ones, there may have been some that were organizational ones, or ones that may have been a mix of multiple. Because "success" is so broadly defined here, there is room for improvement in future projects; we must know what type of success happened in order to more accurately establish their causes and hence will be able to predict certain incidents better than others.

A notable variable that we removed was ransom, which is a binary variable describing whether or not a monetary ransom was demanded during the terrorist incident. Due to the overwhelming number of undefined observations for this variable, we had to remove it from our dataset. The variable nperps, number of perpetrators, had to be removed for a similar reason. Ransom could have played a key role in our classification, since we predicted it to be influential in our second hypothesis for our second research question. However due to the fact we excluded it from our dataset, we were unable to explore the relationship between ransom and extended terrorist incidents. As a result of the large number of missing values of variables from observations before the year 1997, we needed to remove all the prior observations. As a result, we focused only on the most recent decades of terrorist incidents. We therefore missed out on more than 70,000 observations. Therefore the greatest challenge in this project was deciding what to exclude and what to keep during our data cleaning process.

Recommendations

If a similar project is being pursued in the future, it would be best to limit the amount of parameters; for example, we used the target type parameter without narrowing it down, so the results for the naive bayes and classification methods were vastly different. If, instead of using all 20 target types, we used a single target type, it would have yielded results that would make it easier to come to a decision on which model is better for research question 2 specifically.

Our current study is an initial step towards applying machine learning tools towards classifying terrorist incidents. However our study should be interpreted with caution as a result of our imbalanced dataset and lackluster performance in AUC score. For future projects looking to further apply classification models to successful and extended terrorist activities, we recommend implementing some sort of technique to balance the observations used for our research questions. There was an overwhelming majority of successful terrorist incidents to unsuccessful ones, and similarly there was a majority of non-extended terrorist incidents to extended terrorist incidents. Considering that our models had difficulty successfully classifying unsuccessful and extended terrorist incidents, it would make sense to create or record new observations in order to balance the two variables. Having a balanced dataset will remove the problem of our classification models ignoring the minority classes and will therefore do a better job in finding the decision boundary.

Future research could also explore the geographic and attack categories of this project by visualizing incidents and splitting datasets by world region and attack type. Tailoring our classification by running separate models based on world regions and attack types could provide better insight for the specific regions and incident types. For example, running classification models using data exclusively from North America could do a better job in finding relevant predictors than running one large classification using global data. Furthering this project could also entail adding visual components through geospatial data of variables, like showcasing successful attacks by region over time. A deeper understanding of terrorist incidents can be reached by examining specific regions and attack types using the GTD and modeling/visualization.

References

National Consortium for the Study of Terrorism and Responses to Terrorism (START), University of Maryland. (2018). The Global Terrorism Database (GTD) [Data file]. Retrieved from https://www.start.umd.edu/gtd

Global Terrorism Index 2022, Institute for Economics and Peace. (2022). Retreived from https://www.visionofhumanity.org/wp-content/uploads/2022/03/GTI-2022-web-09062022.pdf

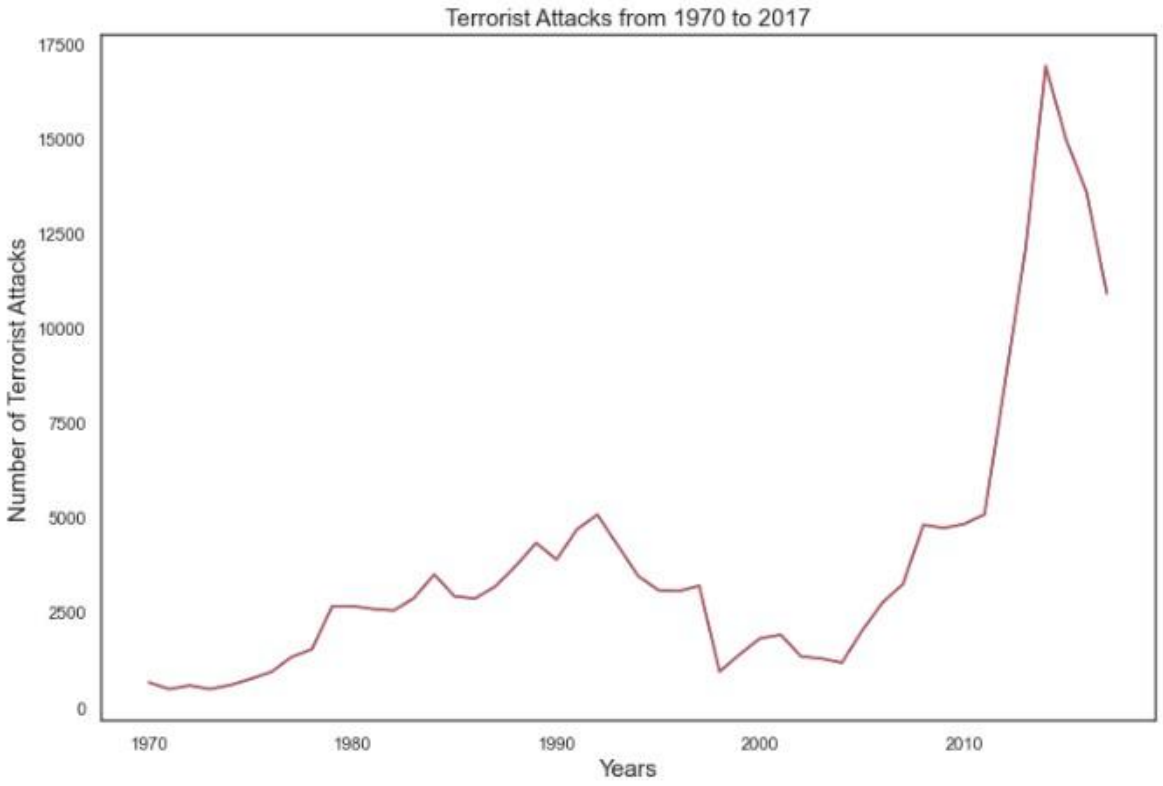
Ibrahim Toure and Aryya Gangopadhyay, "Real time big data analytics for predicting terrorist incidents," 2016 IEEE Symposium on Technologies for Homeland Security (HST), 2016, pp. 1-6, doi: 10.1109/THS.2016.7568906.

Krause, P. (2018). When terrorism works: explaining success and failure across varying targets and objectives. In *When Does Terrorism Work?* (pp. 33-51). Routledge.

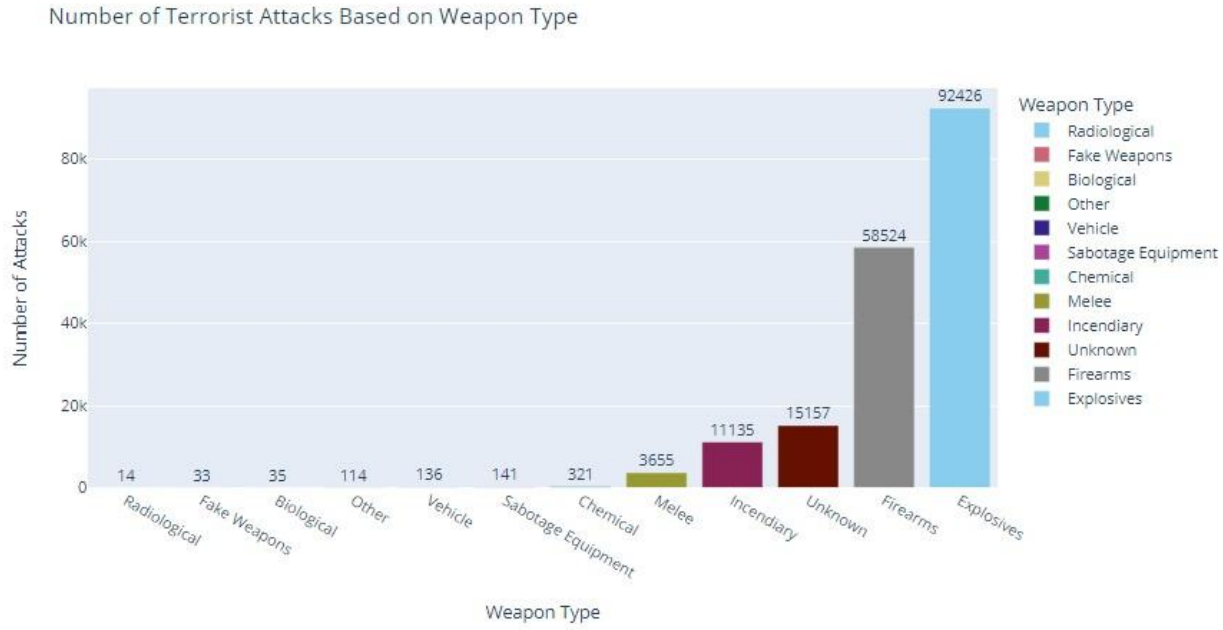
LaFree, G., Dugan, L., & Miller, E. (2014). Putting terrorism in context : Lessons from the global terrorism database. Taylor & Francis Group.

Appendix

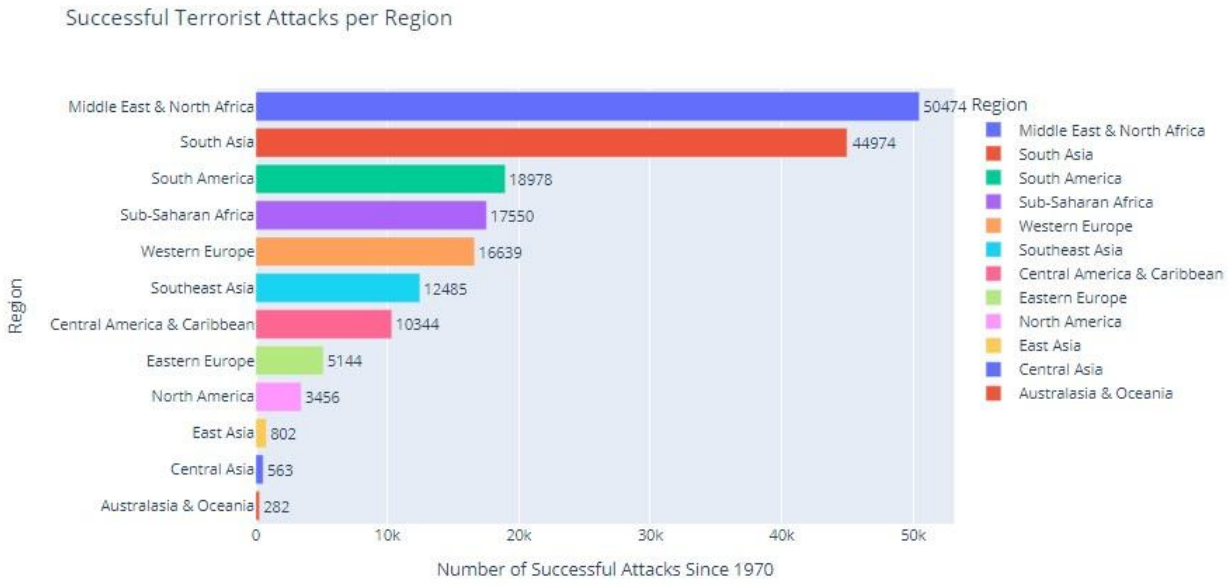
A. “Terrorist Attacks from 1970 to 2017”



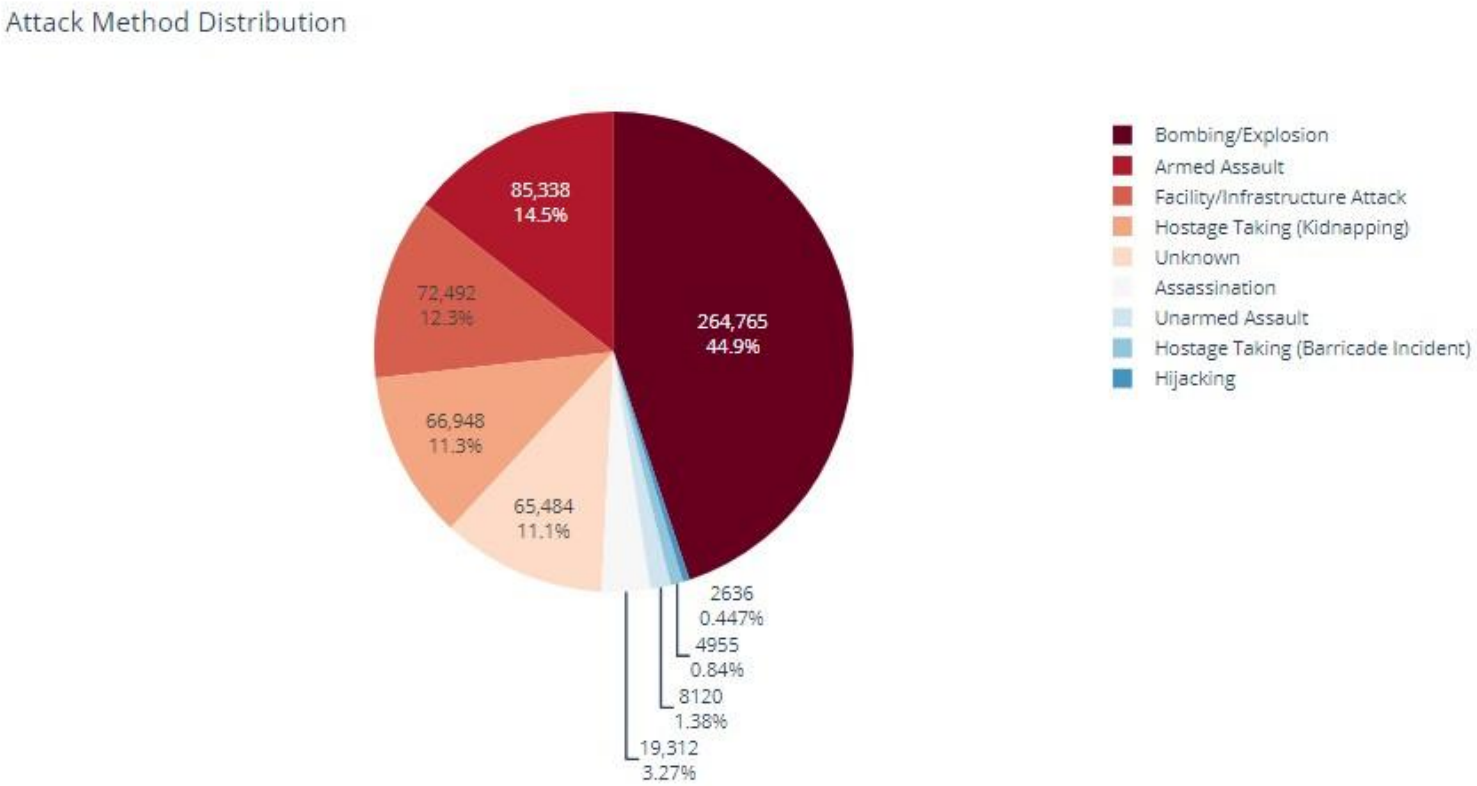
B. “Number of Terrorist Attacks Based on Weapon Type”



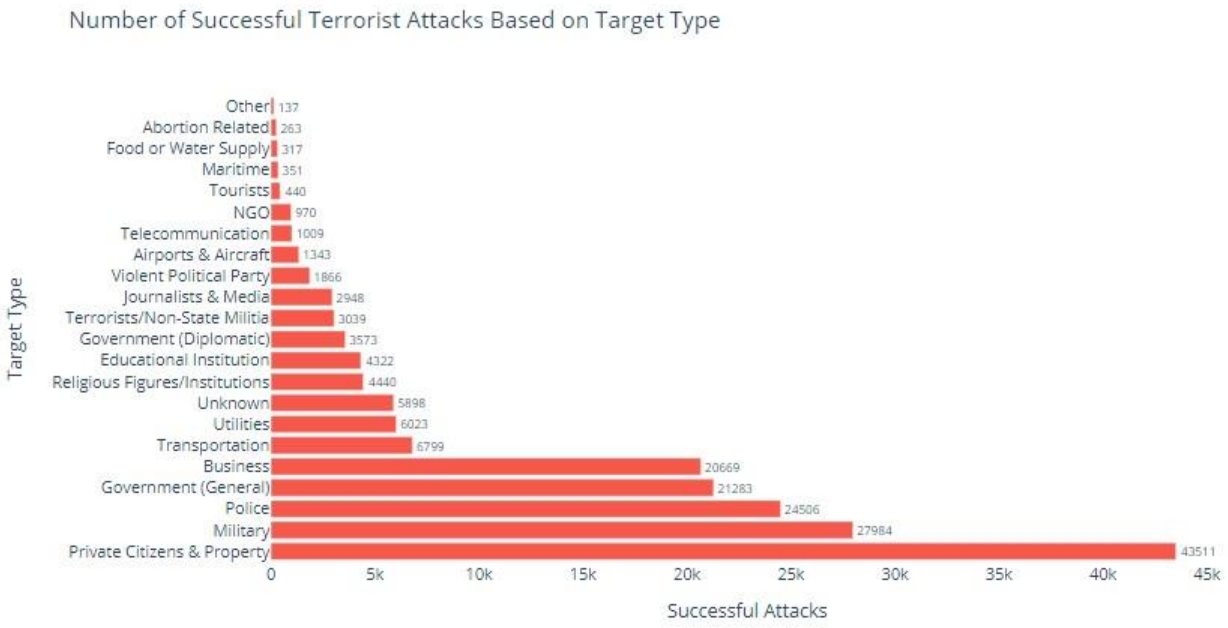
C. “ Successful Terrorist Attacks per Region”



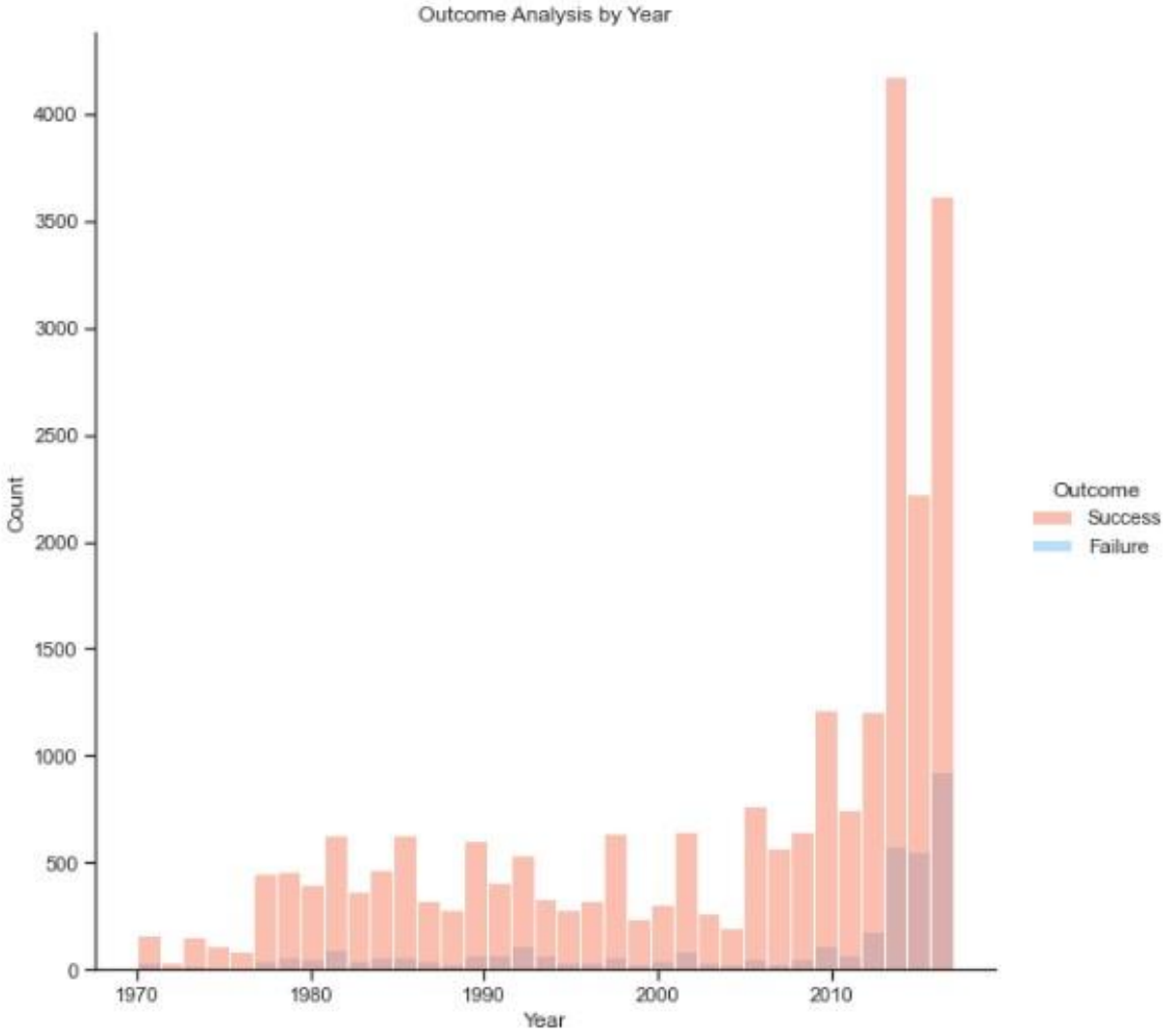
D. “Attack Method Distribution”



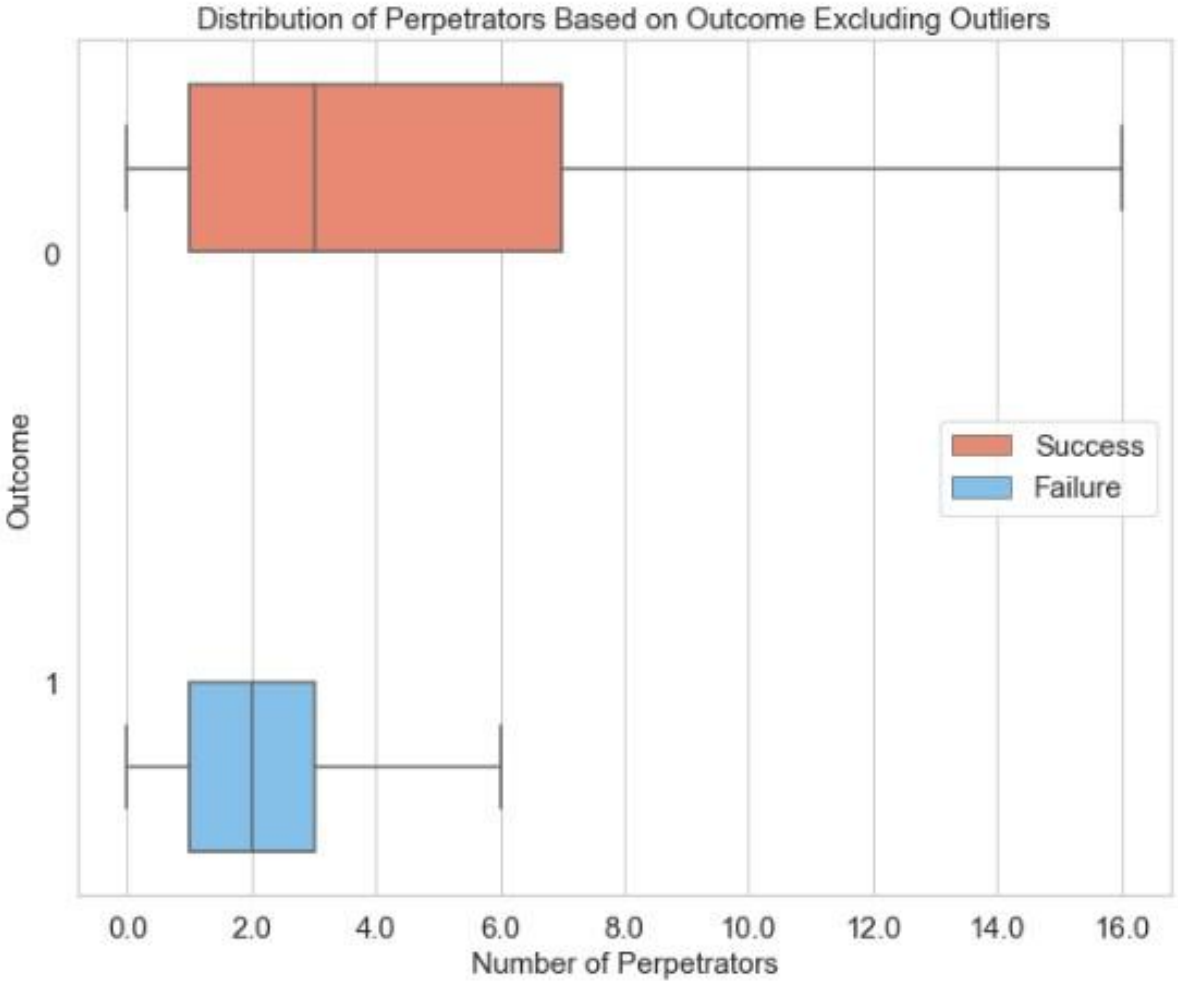
E. “Number of Successful Terrorist Attack Based on Target Type”



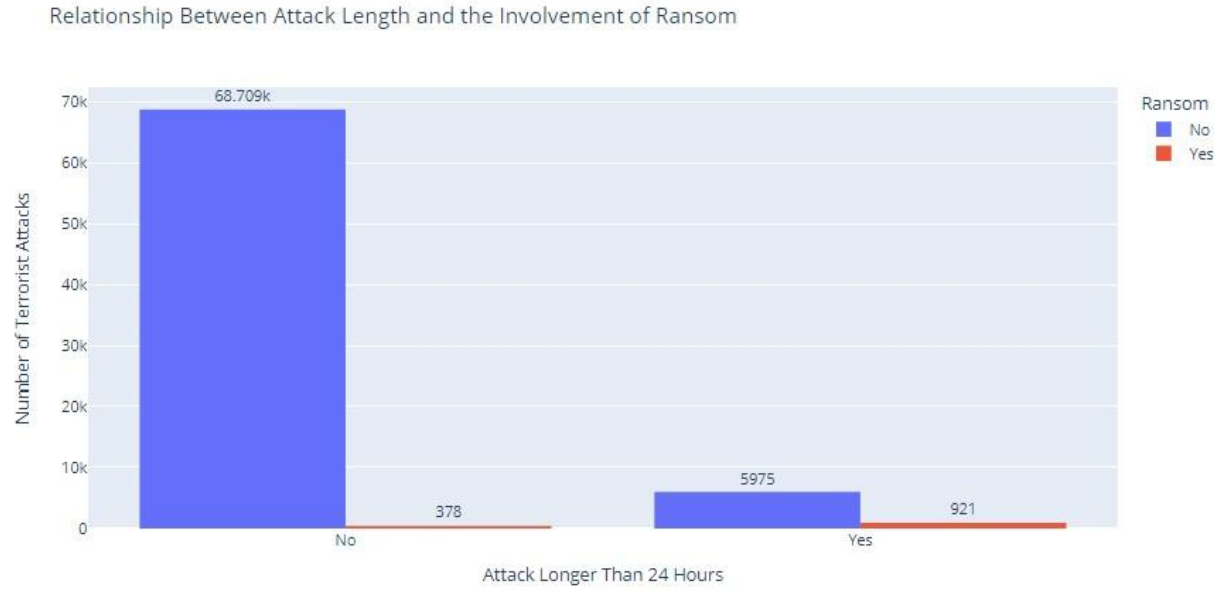
F. “Outcome Analysis Per Year”



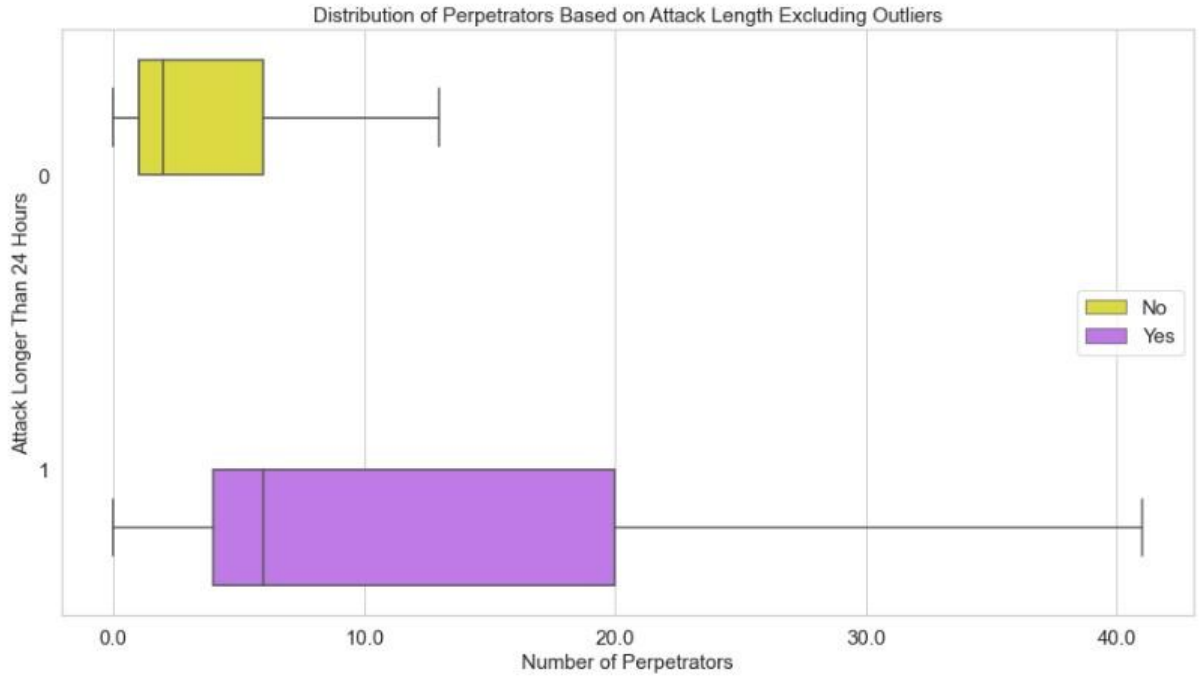
G. “Distribution of Perpetrators Based on Outcome Excluding Outliers”



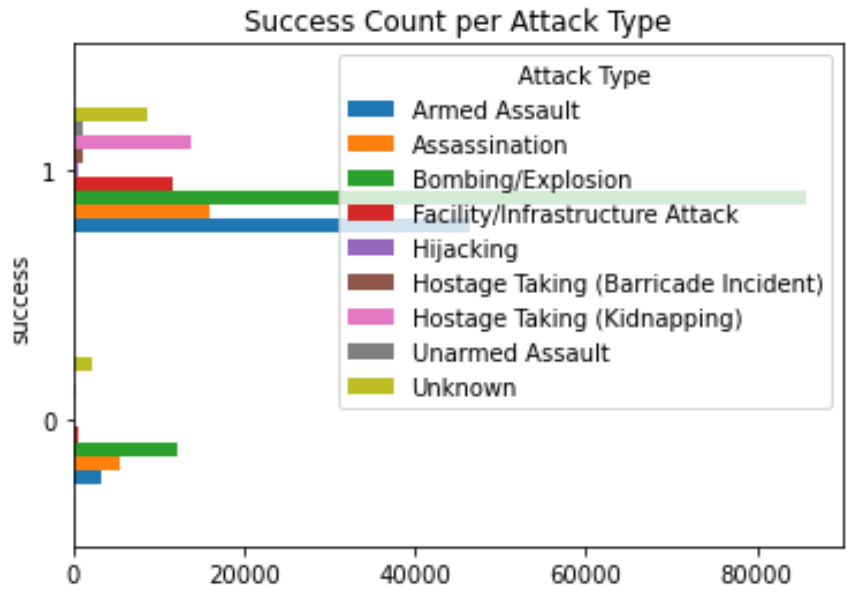
H. “Relationship Between Attack Length and the Involvement of Ransom”



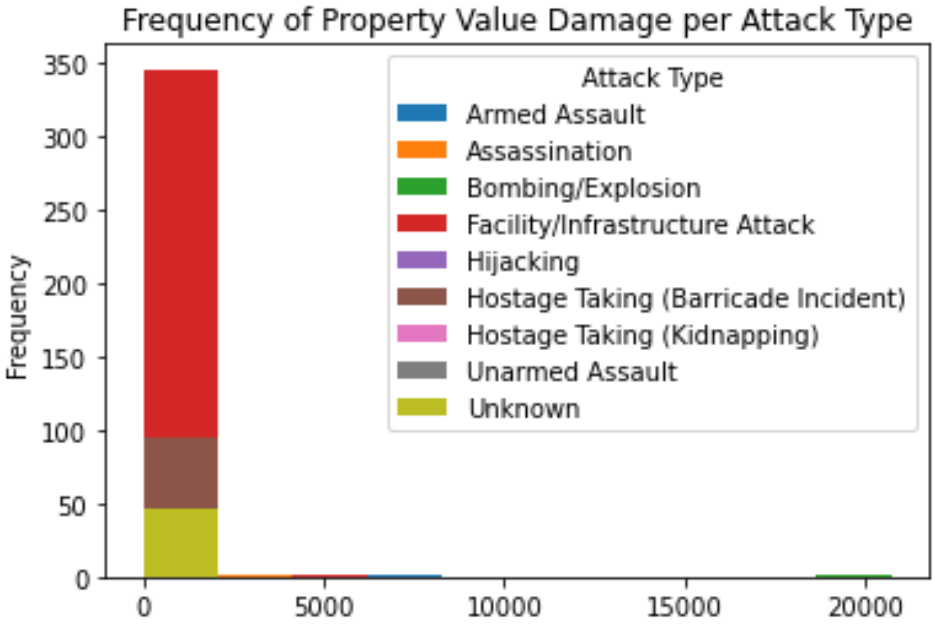
I. “Distribution of Perpetrators Based on Attack Length Excluding Outliers”



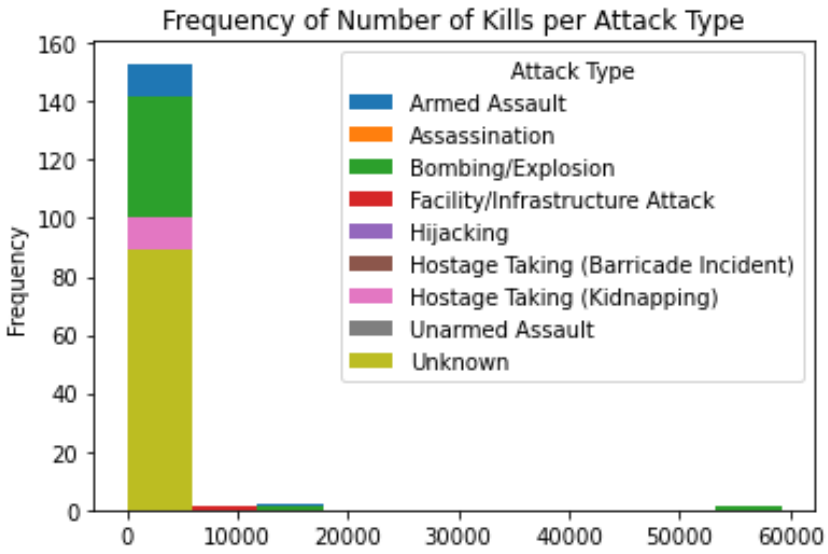
J. "Success Count per Attack Type"



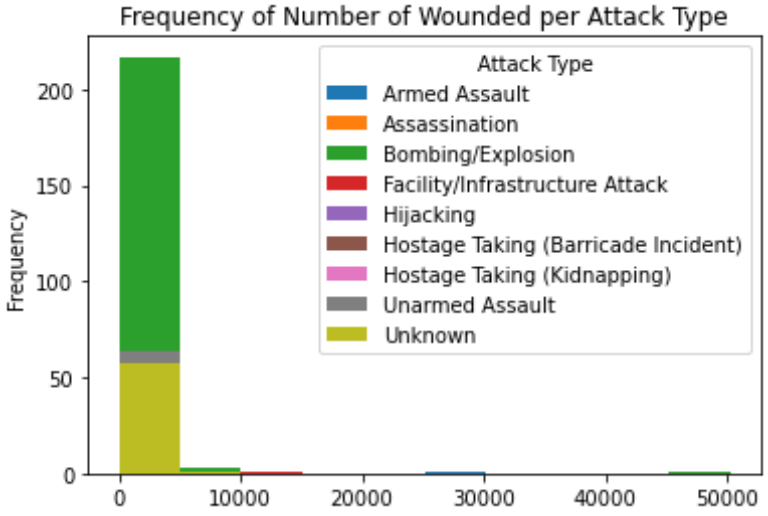
K. "Frequency of Property Value Damage per Attack Type"



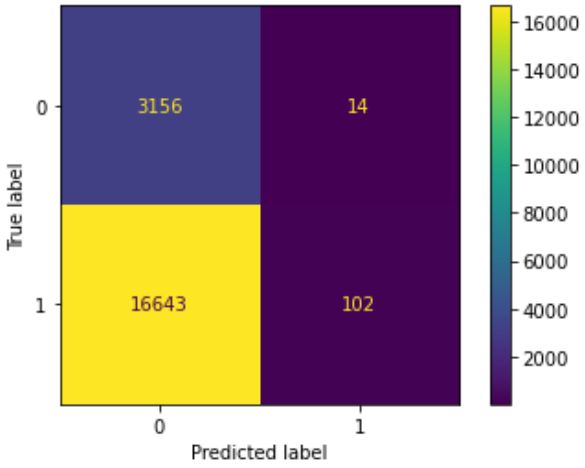
L. "Frequency of Number of Kills per Attack Type"



M. "Frequency of Number of Wounded per Attack Type"



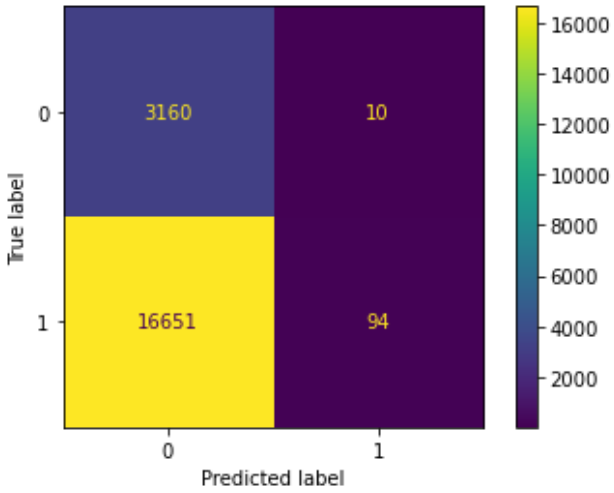
N. Q1 Logistic Regression Confusion Matrix



O. Q1 Logistic Regression Coefficients

variable	coefficient
property	2.0833759602674475
ishostkid	1.6883765623080216
nkill	1.1777989894155847
nkillus	0.8050060706847431
extended	0.5870556329632706
Police	0.5824978819029027
Terrorists/Non-State Militia	0.5778338223048968
Hostage Taking (Kidnapping)	0.5721013179620756
Business	0.481829160717011
Bombing/Explosion	0.38386570775373047

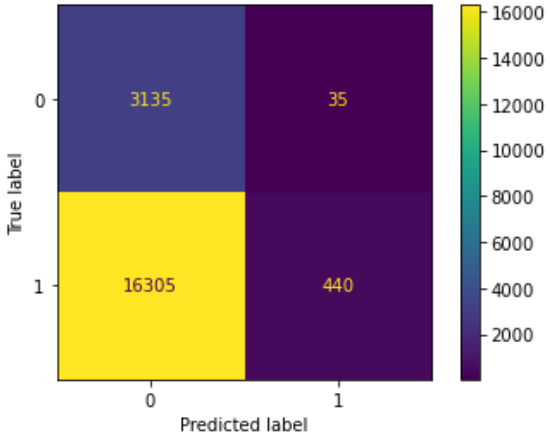
P. Q1 Ridge Regression Confusion Matrix



Q. Q1 Ridge Regression Coefficients

variable	coefficient
property	1.115890730434172
nkill	0.6066090609726278
ishostkid	0.572236087462989
Private Citizens & Property	0.42571667985472567
Police	0.3280974752821491
Bombing/Explosion	0.32258170951672727
Firearms	0.28859834738930823
extended	0.2647971466204123
Terrorists/Non-State Militia	0.2455324483511145
Hostage Taking (Kidnapping)	0.2406512651272702

R. Q1 LASSO Regression Confusion Matrix



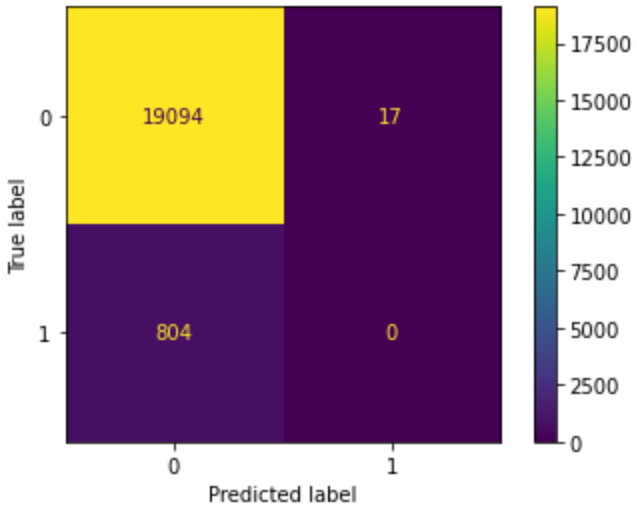
S. Q1 Logistic Regression Coefficients

variable	coefficient
nkill	0.5406573556732588

…

nwoundte	-0.005740030421039889
doubtterr	-0.10855880519506908
nkillter	-0.23912190631609356
crit3	-1.1314605704074538
crit1	-1.458857826998269
intercept	-2.5125903725200556
crit2	-3.5267625935194586

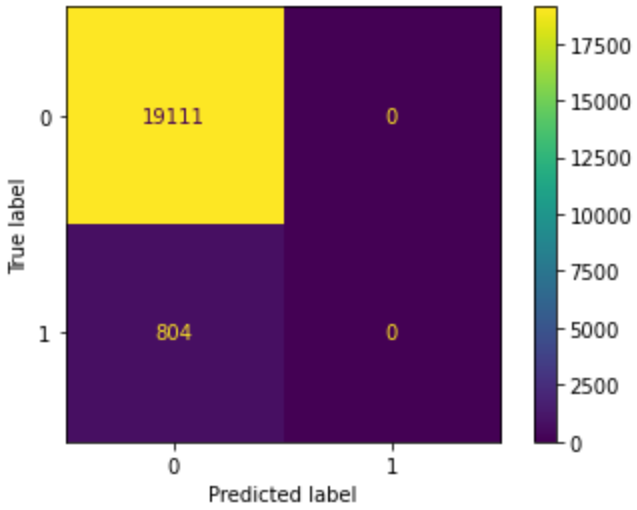
T. Q2 Logistic Regression Confusion Matrix



U. Q2 Logistic Regression Coefficients

variable	coefficient
ishostkid	6.668386143252695
property	2.0618604059227454
claimed	0.9696062810299602
NGO	0.580507005895362
guncertain1	0.3883878184650146
Hijacking	0.3402236933492966
Hostage Taking (Kidnapping)	0.314410691075226
Maritime	0.2713263820525066
Journalists & Media	0.2289419342894803
Central Asia	0.21473885985347368

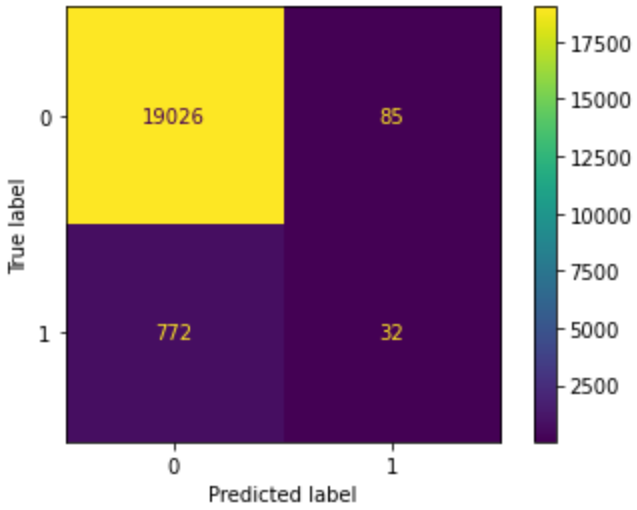
V. Q2 Ridge Regression Confusion Matrix



W. Q2 Logistic Regression Coefficients

variable	coefficient
ishostkid	2.0628684593685263
Hostage Taking (Kidnapping)	0.9071488841925164
success	0.23678203973834286
claimed	0.19330012145514702
NGO	0.17991939702656976
Central Asia	0.16405300294951233
guncertain1	0.135192668014139
Journalists & Media	0.11444524237736636
Business	0.06869000281699562
Maritime	0.0682112764184324

X. Q2 LASSO Regression Confusion Matrix



Y. Q2 Logistic Regression Coefficients

variable	coefficient
ishostkid	0.0852228267304455

…

Bombing/Explosion	-0.038341943286317896
success	-0.11194085037432759
Explosives	-0.2702602245937139
crit3	-0.4787812585194973
crit1	-2.402930136097883
crit2	-3.2867286222818066
intercept	-4.193200880883061

Code

Data Cleaning and Exploratory Data Analysis:

```
# %%
import pandas as pd
import numpy as np

df = pd.read_excel("C:\\Users\\Benja\\Downloads\\globalterrorismdb_0522dist.xlsx")
pd.set_option('display.max_columns', 1000)

#%%
# Getting subset of numerical variables
df_obj = df.select_dtypes(exclude=[np.number])
df_num = df.select_dtypes(include=[np.number])

#extracted_cols = df_obj['region_txt','attacktype1_txt','targtype1_txt','weaptype1_txt']
df_num['region'] = df['region_txt']
df_num['attacktype1'] = df['attacktype1_txt']
df_num['targtype1'] = df['targtype1_txt']
df_num['weaptype1'] = df['weaptype1_txt']


# Heat map (Not used)
#sns.heatmap(df_num.corr());

#%%
# Bar plot used for data exploration
y1 = df.groupby(['Attack Type']).success.value_counts().unstack(0).plot.barh()
y2 = df.groupby(['Attack Type']).propvalue.value_counts().unstack(0).plot.hist()
y3 = df.groupby(['Attack Type']).nkill.value_counts().unstack(0).plot.hist()
y4 = df.groupby(['Attack Type']).nwound.value_counts().unstack(0).plot.hist()

y1.set_title("Success Count per Attack Type")
y2.set_title("Frequency of Property Value Damage per Attack Type")
y3.set_title("Frequency of Number of Kills per Attack Type")
y4.set_title("Frequency of Number of Wounded per Attack Type")

#%%
# Subsetting based on year
column = df_num["iyear"]
count = column[column > 1997].count()

#removing values from before 1998
new_df = df_num.loc[df_num['iyear'] > 1997]
#removing unnecessary columns
final_df = new_df.drop(columns=['iday','country','specificity','targsubtype1',
                                'natlty1','nperps','weapsubtype1','eventid','weaptype4',
                                'weapsubtype4', 'claimmode3', 'guncertain3', 'claim3',
                                'attacktype3','claimmode2','ransompaidus','ransomamtus',
                                'ransompaid','ransomamt','targsubtype3','natlty3','targtype3',
                                'weapsubtype3','weaptype3','guncertain2','claim2','nhours',
                                'compclaim','attacktype2','ndays','weapsubtype2','nreleased',
                                'nhostkidus','hostkidoutcome','nhostkid','weaptype2','ransom',
                                'targsubtype2','natlty2','targtype2','claimmode','alternative',
                                'propvalue','propextent','INT_LOG','INT_IDEO','INT_MISC','INT_ANY'])


#%%
# Dropping all nans
finaldf = final_df.dropna(axis=0)

# dropping all values == -9
finaldf = finaldf[finaldf.property != -9]
finaldf = finaldf[finaldf.ishostkid != -9]
finaldf = finaldf[finaldf.claimed != -9]
finaldf = finaldf[finaldf.nperpcap != -99]
finaldf = finaldf[finaldf.nperpcap != -9]
finaldf = finaldf[finaldf.vicinity != -9]

#%%
#Encoding the four categorical non-binary variables
one_hot = pd.get_dummies(finaldf['region'])
finaldf = finaldf.drop('region',axis = 1)
finaldf = finaldf.join(one_hot)


atkhot = pd.get_dummies(finaldf['attacktype1'])
finaldf = finaldf.drop('attacktype1',axis = 1)
finaldf = finaldf.join(atkhot)

finaldf = finaldf.drop(columns=['Unknown'])

targhot = pd.get_dummies(finaldf['targtype1'])
finaldf = finaldf.drop('targtype1',axis = 1)
finaldf = finaldf.join(targhot)
```

```
finaldf = finaldf.drop(columns=['Unknown'])
finaldf = finaldf.drop(columns=['Other'])

weaphot = pd.get_dummies(finaldf['weaptype1'])
finaldf = finaldf.drop('weaptype1',axis = 1)
finaldf = finaldf.join(weaphot)

finaldf = finaldf.drop(columns=['Unknown'])
finaldf = finaldf.drop(columns=['Other'])

sss = finaldf.isna().sum()

#%%
finaldf.describe()

#%%
finaldf.to_excel("C:\\Users\\Benja\\Downloads\\clean_gtd.xlsx")
```

```
import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
```

```
import plotly.express as px
```

```
#Load the dataset, print the first 10 rows

df = pd.read_csv("C:/Users/andre/Downloads/globalterrorismdb_0718dist.csv/globalterrorismdb_0718dist.csv")
pd.set_option('display.max_columns', 1000)
df.head(10)
```

```
#Include all the parameters we're interested in

target = df[['eventid', 'iyear', 'extended', 'country', 'country_txt', 'region', 'region_txt',
             'success', 'suicide', 'attacktype1', 'attacktype1_txt', 'targtype1', 'targtype1_txt',
             'nperps', 'weaptype1', 'weaptype1_txt', 'nkill', 'ransom', 'ransompaid']]
```

```
#Only eventid, to be used as a counting measure, and the year
dfLine = df[['iyear','eventid']]
perYear = dfLine.groupby('iyear').count().reset_index()

x = perYear['iyear']
y = perYear['eventid'].astype(int)


#Setup the figure sizes, labels, and font sizes
sns.set(rc = {'figure.figsize':(12,8)})
sns.set(style = 'white')

line = sns.lineplot(x = x, y = y)

line.set_xlabel("iyear", fontsize = 15)
line.set_ylabel("eventid", fontsize = 15)
line.set_title('Terrorist Attacks from 1970 to 2018', fontsize = 15)

plt.xlabel('Years')
plt.ylabel('Number of Terrorist Attacks')
plt.plot(x,y,'r', ms=3)
```

```
print(df['weaptype1_txt'].unique())
```

```
['Unknown' 'Explosives' 'Incendiary' 'Firearms' 'Chemical' 'Melee'
 'Sabotage Equipment'
 'Vehicle (not to include vehicle-borne explosives, i.e., car or truck bombs)'
 'Fake Weapons' 'Radiological' 'Other' 'Biological']
```

```
#Number of instances of each weapon used
weapons = df[['weaptype1', 'weaptype1_txt']]
weapon = weapons.groupby(['weaptype1_txt']).count().reset_index()

#Change the long variable name to something shorter
weapon.loc[11,'weaptype1_txt'] = 'Vehicle'

#Sanity check
weapon = weapon.sort_values(by='weaptype1', ascending = True)
weapon
```

```
#Using plotly express, creating a bar graph
fig = px.bar(weapon, x = 'weaptype1_txt', y = 'weaptype1', text = 'weaptype1',
             color = 'weaptype1_txt', color_discrete_sequence=px.colors.qualitative.Safe,
             title = 'Number of Terrorist Attacks Based on Weapon Type',
             labels = {'weaptype1_txt':'Weapon Type'})
fig.update_traces(textposition = 'outside')
fig.update_xaxes(title_text = 'Weapon Type')
fig.update_yaxes(title_text = 'Number of Attacks')
fig.show()
```

```
region = target[['region_txt', 'success']]

#Successful attacks per region
groupRegion = region.groupby(['region_txt']).count().reset_index()
groupRegionNew = groupRegion.sort_values(by='success', ascending = False)
```

```
#Bar plot using plotly express
fig2 = px.bar(groupRegionNew, orientation = 'h', y = 'region_txt', x = 'success',
              color = 'region_txt', text = 'success', labels={'region_txt':'Region'},
             title = 'Successful Terrorist Attacks per Region')
fig2.update_yaxes(title_text = 'Region')
fig2.update_xaxes(title_text = 'Number of Successful Attacks Since 1970')
fig2.update_traces(textposition = 'outside', cliponaxis=False)
fig2.show()
```

```
attackType = df[['attacktype1','attacktype1_txt']]
```

```
#Checking if variable names are not too long
print(df['attacktype1_txt'].unique())
```

```
['Assassination' 'Hostage Taking (Kidnapping)' 'Bombing/Explosion'
 'Facility/Infrastructure Attack' 'Armed Assault' 'Hijacking' 'Unknown'
 'Unarmed Assault' 'Hostage Taking (Barricade Incident)']
```

```
#Creating a pie chart with plotly express
fig3 = px.pie(attackType, values = 'attacktype1', names = 'attacktype1_txt',
              color_discrete_sequence=px.colors.sequential.RdBu, title = "Attack Method Distribution")
fig3.update_traces(textinfo='percent+value')
fig3.show()
```

```
#Successes distinguished by target type
targetType = target[['targtype1_txt','success']]
targetType = targetType.groupby(['targtype1_txt']).count().reset_index()
targetType = targetType.sort_values(by='success', ascending = False)
```

```
fig4 = px.bar(targetType, orientation = 'h', x = 'success', y = 'targtype1_txt', text = 'success',
             title = 'Number of Successful Terrorist Attacks Based on Target Type',
             labels = {'targtype1_txt':'Target Type'})

#Changing the bar colors along with making the counts visible outside
fig4.update_traces(marker_color = 'rgb(164, 129, 230)', textposition = 'outside', cliponaxis = False)

fig4.update_yaxes(title_text = 'Target Type')
fig4.update_xaxes(title_text = 'Successful Attacks')
fig4.show()
```

```
perp = target[['success','nperps','iyear']]
perp = perp.dropna()

knownPerp = perp[perp.nperps >= 0]
knownPerp = knownPerp.rename(columns={'success' : 'Outcome'})

knownPerp = knownPerp.replace({'Outcome' : {0 : 'Failure', 1 : 'Success'}})
knownPerp
```

```
#Creating custom color pallettes with Seaborn
sns.set(rc = {'figure.figsize':(10,8)})
custom = ['#FB7E5F','#73C3FA']
unique = sns.set_palette(sns.color_palette(custom))

#Creating a histogram overlay with successes and failures per year
sns.set_theme(style = "ticks", palette = unique)
sns.displot(data=knownPerp, x = 'iyear', stat = 'count', hue = 'Outcome',
            height = 8)
plt.xlabel('Year')
plt.title('Outcome Analysis by Year')
plt.show()
```

```
groupSuccess = target[['success','nperps']]
groupSuccess = groupSuccess.dropna()

#values of -9 are frequently seen
groupSuccess = groupSuccess[perp.nperps >= 0]

#Rename success variable, change the binary results to "failure" and "success"
groupSuccess = groupSuccess.rename(columns={'success' : 'Outcome'})
groupSuccess = groupSuccess.replace({'Outcome' : {0 : 'Failure', 1 : 'Success'}})
```

```
#Boxplot, perpetrators vs. outcome, without outliers
sns.set_theme(style = 'whitegrid')
outcomePlot = sns.boxplot(data = groupSuccess, y = 'Outcome', x = 'nperps', hue = 'Outcome', palette = ['#FB7E5F','#73C3FA'],
            showfliers = False)

outcomePlot.set_xticklabels(outcomePlot.get_xticks(), size = 15)
outcomePlot.set_yticklabels(outcomePlot.get_yticks(), size = 15)


outcomePlot.set_xlabel("nperps", fontsize = 15)
outcomePlot.set_ylabel("Outcome", fontsize = 15)
outcomePlot.set_title('Distribution of Perpetrators Based on Outcome Excluding Outliers', fontsize = 15)

plt.xlabel('Number of Perpetrators')
plt.legend(fontsize = 15, loc='center right')
```

```
ransomExt = target[['extended', 'ransom']]
ransomExt = ransomExt.dropna()

#Some values are -9
ransomExt = ransomExt[ransomExt.ransom >= 0]

#Replace binary counts with Yes or No
ransomExt = ransomExt.rename(columns={'extended' : 'Extended', 'ransom' : 'Ransom'})
ransomExt = ransomExt.replace({'Extended' : {0 : 'No', 1 : 'Yes'}})
ransomExt = ransomExt.replace({'Ransom' : {0 : 'No', 1 : 'Yes'}})

#Create an empty column to append the counts
i = 0
column = []
while i < 75983:
    column.append(i)
    i = i + 1

ransomExt['Count'] = column

#Every row needs a count of 1
ransomExt.loc[ransomExt['Extended'] == 'Yes', 'Count'] = 1
ransomExt.loc[ransomExt['Extended'] != 'Yes', 'Count'] = 1

ransomExt
```

```
#Comparative bar chart
fig5 = px.histogram(ransomExt, x = 'Extended', barmode = 'group', histfunc='count', color = 'Ransom',
                    text_auto = True, title = 'Relationship Between Attack Length and the Involvement of Ransom')
fig5.update_traces(textposition = 'outside')
fig5.update_xaxes(title_text = 'Attack Longer Than 24 Hours')
fig5.update_yaxes(title_text = 'Number of Terrorist Attacks')
fig5.show()
```

```
#Setup for the final boxplot
extPerp = target[['extended', 'nperps']]
extPerp = extPerp.dropna()
extPerp = extPerp[extPerp['nperps'] >= 0]
extPerp = extPerp.replace({'extended' : {0 : 'No', 1 : 'Yes'}})
```

```
#Seaborn boxplot without outliers
sns.set(rc = {'figure.figsize':(15,8)})
sns.set_theme(style = 'whitegrid')
boxPlot = sns.boxplot(data = extPerp, y = 'extended', x = 'nperps', hue = 'extended', palette = ['#F6F329','#C368FA'],
           showfliers = False)

boxPlot.set_xticklabels(boxPlot.get_xticks(), size = 15)
boxPlot.set_yticklabels(boxPlot.get_yticks(), size = 15)


boxPlot.set_xlabel("nperps", fontsize = 15)
boxPlot.set_ylabel("extended", fontsize = 15)
boxPlot.set_title('Distribution of Perpetrators Based on Attack Length Excluding Outliers', fontsize = 15)

plt.xlabel('Number of Perpetrators')
plt.ylabel('Attack Longer Than 24 Hours')
plt.legend(fontsize = 15, loc='center right')
```

Classification Modeling:

```
# %%
import pandas as pd
import numpy as np
from numpy import mean
from numpy import std
from sklearn.datasets import make_classification
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import log_loss, roc_auc_score, recall_score, precision_score, average_precision_score, f1_score, cl


df = pd.read_excel("C:\\Users\\Benja\\Downloads\\clean_gtd.xlsx")
pd.set_option('display.max_columns', 1000)
df = df.iloc[: , 1:]
df = df[df.nperpcap != -9]

#%%
#Logistic Regression Attempt
numeric_cols = ['iyear', 'imonth', 'latitude', 'longitude', 'nperpcap','nkill','nkillus','nkillter','nwound']
cat_cols = list(set(df.columns) - set(numeric_cols) - {'target'})

cat_cols.sort()

df_train, df_test = train_test_split(df, test_size=0.2, random_state=1, stratify=df['success'])

#y = df_train['success']
#X = df_train

scaler = StandardScaler()
scaler.fit(df_train[numeric_cols])

def get_features_and_target_arrays(df, numeric_cols, cat_cols, scaler):
    X_numeric_scaled = scaler.transform(df[numeric_cols])
    X_categorical = df[cat_cols].to_numpy()
    X = np.hstack((X_categorical, X_numeric_scaled))
    y = df['success']
    return X, y

X, y = get_features_and_target_arrays(df_train, numeric_cols, cat_cols, scaler)

clf = LogisticRegression(penalty='none', max_iter = 1000)
clf.fit(X, y)

y_test = df_test['success']
X_test = df_test

#%%
# Model Evaluation
plot_roc_curve(clf, X_test, y_test)

plot_precision_recall_curve(clf, X_test, y_test)
```

```
test_prob = clf.predict_proba(X_test)[:, 1]
test_pred = clf.predict(X_test)

print('Confusion Matrix')
plot_confusion_matrix(clf, X_test, y_test)

coefficients = np.hstack((clf.intercept_, clf.coef_[0]))
coo = pd.DataFrame(data={'variable': ['intercept'] + cat_cols + numeric_cols, 'coefficient': coefficients})

pd.DataFrame(data={'variable': numeric_cols, 'unit': np.sqrt(scaler.var_)})
```

```
#%%
#Ridge Regression Attempt
numeric_cols = ['iyear', 'imonth', 'latitude', 'longitude', 'nperpcap','nkill','nkillus','nkillter','nwound']
cat_cols = list(set(df.columns) - set(numeric_cols) - {'target'})

cat_cols.sort()

df_train, df_test = train_test_split(df, test_size=0.2, random_state=1, stratify=df['success'])

#y = df_train['success']
#X = df_train

scaler = StandardScaler()
scaler.fit(df_train[numeric_cols])

def get_features_and_target_arrays(df, numeric_cols, cat_cols, scaler):
    X_numeric_scaled = scaler.transform(df[numeric_cols])
    X_categorical = df[cat_cols].to_numpy()
    X = np.hstack((X_categorical, X_numeric_scaled))
    y = df['success']
    return X, y

X, y = get_features_and_target_arrays(df_train, numeric_cols, cat_cols, scaler)

clf = LogisticRegression(penalty='l2', max_iter = 1000)
clf.fit(X, y)

y_test = df_test['success']
X_test = df_test
```

```
#%%
# Model Evaluation
plot_roc_curve(clf, X_test, y_test)

plot_precision_recall_curve(clf, X_test, y_test)

test_prob = clf.predict_proba(X_test)[:, 1]
test_pred = clf.predict(X_test)

print('Confusion Matrix')
plot_confusion_matrix(clf, X_test, y_test)

coefficients = np.hstack((clf.intercept_, clf.coef_[0]))
coo2 = pd.DataFrame(data={'variable': ['intercept'] + cat_cols + numeric_cols, 'coefficient': coefficients})
```

```
pd.DataFrame(data={'variable': numeric_cols, 'unit': np.sqrt(scaler.var_)})
```

```
#%%
#LASSO Regression Attempt
numeric_cols = ['iyear', 'imonth', 'latitude', 'longitude', 'nperpcap','nkill','nkillus','nkillter','nwound']
cat_cols = list(set(df.columns) - set(numeric_cols) - {'target'})

cat_cols.sort()

df_train, df_test = train_test_split(df, test_size=0.2, random_state=1, stratify=df['success'])

#y = df_train['success']
#X = df_train

scaler = StandardScaler()
scaler.fit(df_train[numeric_cols])

def get_features_and_target_arrays(df, numeric_cols, cat_cols, scaler):
    X_numeric_scaled = scaler.transform(df[numeric_cols])
    X_categorical = df[cat_cols].to_numpy()
    X = np.hstack((X_categorical, X_numeric_scaled))
    y = df['success']
    return X, y

X, y = get_features_and_target_arrays(df_train, numeric_cols, cat_cols, scaler)

clf = LogisticRegression(penalty='l1', solver='liblinear', max_iter = 1000)
clf.fit(X, y)

y_test = df_test['success']
X_test = df_test
```

```
#%%
# Model Evaluation
plot_roc_curve(clf, X_test, y_test)

plot_precision_recall_curve(clf, X_test, y_test)

test_prob = clf.predict_proba(X_test)[:, 1]
test_pred = clf.predict(X_test)

print('Confusion Matrix')
plot_confusion_matrix(clf, X_test, y_test)

coefficients = np.hstack((clf.intercept_, clf.coef_[0]))
coo3 = pd.DataFrame(data={'variable': ['intercept'] + cat_cols + numeric_cols, 'coefficient': coefficients})

pd.DataFrame(data={'variable': numeric_cols, 'unit': np.sqrt(scaler.var_)})
```

```
#%%% QUESTION TWO
#Logistic Regression Attempt
numeric_cols = ['iyear', 'imonth', 'latitude', 'longitude', 'nperpcap','nkill','nkillus','nkillter','nwound']
cat_cols = list(set(df.columns) - set(numeric_cols) - {'target'})

cat_cols.sort()

df_train, df_test = train_test_split(df, test_size=0.2, random_state=1, stratify=df['extended'])

#y = df_train['success']
#X = df_train

scaler = StandardScaler()
scaler.fit(df_train[numeric_cols])

def get_features_and_target_arrays(df, numeric_cols, cat_cols, scaler):
    X_numeric_scaled = scaler.transform(df[numeric_cols])
    X_categorical = df[cat_cols].to_numpy()
    X = np.hstack((X_categorical, X_numeric_scaled))
    y = df['extended']
    return X, y

X, y = get_features_and_target_arrays(df_train, numeric_cols, cat_cols, scaler)

clf = LogisticRegression(penalty='none', max_iter = 1000)
clf.fit(X, y)

y_test = df_test['extended']
X_test = df_test

#%%
# Model Evaluation
plot_roc_curve(clf, X_test, y_test)

plot_precision_recall_curve(clf, X_test, y_test)

test_prob = clf.predict_proba(X_test)[:, 1]
test_pred = clf.predict(X_test)

print('Confusion Matrix')
plot_confusion_matrix(clf, X_test, y_test)

coefficients = np.hstack((clf.intercept_, clf.coef_[0]))
coo = pd.DataFrame(data={'variable': ['intercept'] + cat_cols + numeric_cols, 'coefficient': coefficients})

pd.DataFrame(data={'variable': numeric_cols, 'unit': np.sqrt(scaler.var_)})

#%%
#Ridge Regression Attempt
numeric_cols = ['iyear', 'imonth', 'latitude', 'longitude', 'nperpcap','nkill','nkillus','nkillter','nwound']
cat_cols = list(set(df.columns) - set(numeric_cols) - {'target'})

cat_cols.sort()

df_train, df_test = train_test_split(df, test_size=0.2, random_state=1, stratify=df['extended'])
```

```
#y = df_train['success']
#X = df_train

scaler = StandardScaler()
scaler.fit(df_train[numeric_cols])

def get_features_and_target_arrays(df, numeric_cols, cat_cols, scaler):
    X_numeric_scaled = scaler.transform(df[numeric_cols])
    X_categorical = df[cat_cols].to_numpy()
    X = np.hstack((X_categorical, X_numeric_scaled))
    y = df['extended']
    return X, y

X, y = get_features_and_target_arrays(df_train, numeric_cols, cat_cols, scaler)

clf = LogisticRegression(penalty='l2', max_iter = 1000)
clf.fit(X, y)

y_test = df_test['extended']
X_test = df_test

#%%
# Model Evaluation
plot_roc_curve(clf, X_test, y_test)

plot_precision_recall_curve(clf, X_test, y_test)

test_prob = clf.predict_proba(X_test)[:, 1]
test_pred = clf.predict(X_test)

print('Confusion Matrix')
plot_confusion_matrix(clf, X_test, y_test)

coefficients = np.hstack((clf.intercept_, clf.coef_[0]))
coo2 = pd.DataFrame(data={'variable': ['intercept'] + cat_cols + numeric_cols, 'coefficient': coefficients})

pd.DataFrame(data={'variable': numeric_cols, 'unit': np.sqrt(scaler.var_)})

#%%
#LASSO Regression Attempt
numeric_cols = ['iyear', 'imonth', 'latitude', 'longitude', 'nperpcap','nkill','nkillus','nkillter','nwound']
cat_cols = list(set(df.columns) - set(numeric_cols) - {'target'})

cat_cols.sort()

df_train, df_test = train_test_split(df, test_size=0.2, random_state=1, stratify=df['extended'])

#y = df_train['extended']
#X = df_train

scaler = StandardScaler()
scaler.fit(df_train[numeric_cols])
```

```
def get_features_and_target_arrays(df, numeric_cols, cat_cols, scaler):
    X_numeric_scaled = scaler.transform(df[numeric_cols])
    X_categorical = df[cat_cols].to_numpy()
    X = np.hstack((X_categorical, X_numeric_scaled))
    y = df['extended']
    return X, y

X, y = get_features_and_target_arrays(df_train, numeric_cols, cat_cols, scaler)

clf = LogisticRegression(penalty='l1', solver='liblinear', max_iter = 1000)
clf.fit(X, y)

y_test = df_test['extended']
X_test = df_test

#%%
# Model Evaluation
plot_roc_curve(clf, X_test, y_test)

plot_precision_recall_curve(clf, X_test, y_test)

test_prob = clf.predict_proba(X_test)[:, 1]
test_pred = clf.predict(X_test)

print('Confusion Matrix')
plot_confusion_matrix(clf, X_test, y_test)

coefficients = np.hstack((clf.intercept_, clf.coef_[0]))
coo3 = pd.DataFrame(data={'variable': ['intercept'] + cat_cols + numeric_cols, 'coefficient': coefficients})

pd.DataFrame(data={'variable': numeric_cols, 'unit': np.sqrt(scaler.var_)})
```

```
#%%
# Additional Logistic Regression for Q1 and Q2
#Q1 nkill
X_train, X_test, y_train, y_test = train_test_split(df['nkill'], df['success'], test_size=0.2, random_state=16)
X_test = X_test.array.reshape(-1,1)
X_train = X_train.array.reshape(-1,1)

clf = LogisticRegression(max_iter = 1000)
clf.fit(X_train,y_train)


#%%
# Model Evaluation
plot_roc_curve(clf, X_test, y_test)

plot_precision_recall_curve(clf, X_test, y_test)

test_prob = clf.predict_proba(X_test)[:, 1]
test_pred = clf.predict(X_test)

print('Confusion Matrix')
plot_confusion_matrix(clf, X_test, y_test)
#%%
# Additional Logistic Regression for Q1 and Q2
#Q1 ishostkid
X_train, X_test, y_train, y_test = train_test_split(df['ishostkid'], df['success'], test_size=0.2, random_state=16)
X_test = X_test.array.reshape(-1,1)
X_train = X_train.array.reshape(-1,1)

clf = LogisticRegression(max_iter = 1000)
clf.fit(X_train,y_train)


#%%
# Model Evaluation
plot_roc_curve(clf, X_test, y_test)

plot_precision_recall_curve(clf, X_test, y_test)

test_prob = clf.predict_proba(X_test)[:, 1]
test_pred = clf.predict(X_test)

print('Confusion Matrix')
plot_confusion_matrix(clf, X_test, y_test)


#%%
# Additional Logistic Regression for Q1 and Q2
#Q2 ishostkid
X_train, X_test, y_train, y_test = train_test_split(df['ishostkid'], df['extended'], test_size=0.2, random_state=16)
X_test = X_test.array.reshape(-1,1)
X_train = X_train.array.reshape(-1,1)

clf = LogisticRegression(max_iter = 1000)
clf.fit(X_train,y_train)
```

```
# Model Evaluation
plot_roc_curve(clf, X_test, y_test)
plot_precision_recall_curve(clf, X_test, y_test)
test_prob = clf.predict_proba(X_test)[:, 1]
test_pred = clf.predict(X_test)

print('Confusion Matrix')
plot_confusion_matrix(clf, X_test, y_test)
#%%
# Additional Logistic Regression for Q1 and Q2
#Q2 property
X_train, X_test, y_train, y_test = train_test_split(df['property'], df['extended'], test_size=0.2, random_state=16)
X_test = X_test.array.reshape(-1,1)
X_train = X_train.array.reshape(-1,1)
clf = LogisticRegression(max_iter = 1000)
clf.fit(X_train,y_train)


#%%
# Model Evaluation
plot_roc_curve(clf, X_test, y_test)
plot_precision_recall_curve(clf, X_test, y_test)
test_prob = clf.predict_proba(X_test)[:, 1]
test_pred = clf.predict(X_test)

print('Confusion Matrix')
plot_confusion_matrix(clf, X_test, y_test)
#%%
# Additional Logistic Regression for Q1 and Q2
#Q2 claimed
X_train, X_test, y_train, y_test = train_test_split(df['claimed'], df['extended'], test_size=0.2, random_state=16)
X_test = X_test.array.reshape(-1,1)
X_train = X_train.array.reshape(-1,1)
clf = LogisticRegression(max_iter = 1000)
clf.fit(X_train,y_train)


#%%
# Model Evaluation
plot_roc_curve(clf, X_test, y_test)
plot_precision_recall_curve(clf, X_test, y_test)
test_prob = clf.predict_proba(X_test)[:, 1]
test_pred = clf.predict(X_test)

print('Confusion Matrix')
plot_confusion_matrix(clf, X_test, y_test)
```

```
#%%
# Additional Logistic Regression for Q1 and Q2
#Q2 success
X_train, X_test, y_train, y_test = train_test_split(df['success'], df['extended'], test_size=0.2, random_state=16)
X_test = X_test.array.reshape(-1,1)
X_train = X_train.array.reshape(-1,1)
clf = LogisticRegression(max_iter = 1000)
clf.fit(X_train,y_train)


#%%
# Model Evaluation
plot_roc_curve(clf, X_test, y_test)
plot_precision_recall_curve(clf, X_test, y_test)
test_prob = clf.predict_proba(X_test)[:, 1]
test_pred = clf.predict(X_test)
print('Confusion Matrix')
plot_confusion_matrix(clf, X_test, y_test)

```

Naive Bayes:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.preprocessing import OrdinalEncoder # for encoding categorical features from strings to number arrays
from sklearn.naive_bayes import MultinomialNB, CategoricalNB

from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
```

```
#Load the data
df = pd.read_csv("C:/Users/andre/Downloads/clean_gtd.xlsx - Sheet1.csv")
df = df[df.nperpcap != -9]
df.head(10)
```

```
#Used to see the names of all columns to see which to remove
column_names = df.columns.values.tolist()
```

```
#Interested in extended(Y), method of attack, what they attacked, region, weapon type, drop the rest
df2 = df.drop(df.columns[[0,1,2,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25]], axis = 1)
df2.head(10)
```

```
#Interested in success, weapon type, number slain/killed, longer than 24 hours, number woundedS, method of attack

df = df.drop(df.columns[[0,1,2,4,5,6,7,8,9,10,11,13,14,15,16,17,19,20,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,46,47,48,4
```

```
#Sanity check
df.isnull().sum(axis=0)
```

```
df['success'].value_counts(normalize=True)
```

```
1    0.840817
0    0.159183
Name: success, dtype: float64
```

```
#Select a random seed, randomize the rows for the training and testing data
np.random.seed(9037)
randomization = df.sample(frac=1)

trainsize = round(len(randomization)*0.7)

training_set = randomization[:trainsize].reset_index(drop=True)
test_set = randomization[trainsize:].reset_index(drop=True)
```

```
#See if the training and testing sets are rougly equal to the success value counts of the original data
training_set['success'].value_counts(normalize=True)
```

```
1    0.840674
0    0.159326
Name: success, dtype: float64
```

```
test_set['success'].value_counts(normalize=True)
```

```
1    0.84115
0    0.15885
Name: success, dtype: float64
```

```
#X is every variable except for the success column, Y is the Success column only
trainX = training_set.drop(training_set.columns[[1]], axis = 1)
trainY = training_set['success']
```

```
testX = test_set.drop(training_set.columns[[1]], axis = 1)
testY = test_set['success']
```

```
#Encode the variables, base model off of trained data
from sklearn.preprocessing import LabelEncoder
enc = OrdinalEncoder()
le = LabelEncoder()
trainBrnli = le.fit_transform(trainY)

model = CategoricalNB()
model.fit(trainX,trainBrnli)
```

```
CategoricalNB()
```

```
#Create the testing variables and the predicted values for yhat
testBrnli = le.fit_transform(testY)

colnames = testX.columns
testX = enc.fit_transform(testX)
testX = pd.DataFrame(testX, columns=colnames)

yhattest = model.predict(testX)
```

```
confM = pd.crosstab(yhattest, testY)
confM
```

success	0	1
row_0		
0	2649	3904
1	2096	21222

```
#Accuracy of the model
accuracy_score(yhattest, testBrnli)
```

```
0.7991362860299287
```

```
#Ratios to compare if the training and testing sets will be roughly the same later
df2['extended'].value_counts(normalize=True)
```

```
0    0.959607
1    0.040393
Name: extended, dtype: float64
```

```
#Randomize and create testing and training sets
np.random.seed(3944)
randomization = df2.sample(frac=1)

trainsize = round(len(randomization)*0.7)

training_set = randomization[:trainsize].reset_index(drop=True)
test_set = randomization[trainsize:].reset_index(drop=True)
```

```
training_set['extended'].value_counts(normalize=True)
```

```
0    0.959527
1    0.040473
Name: extended, dtype: float64
```

```
trainX = training_set.drop(training_set.columns[[0]], axis = 1)
trainY = training_set['extended']
```

```
testX = test_set.drop(training_set.columns[[0]], axis = 1)
testY = test_set['extended']
```

```
#Same process as previously
enc = OrdinalEncoder()
le = LabelEncoder()
trainBrnli = le.fit_transform(trainY)

model = CategoricalNB()
model.fit(trainX,trainBrnli)
```

```
testBrnli = le.fit_transform(testY)

colnames = testX.columns
testX = enc.fit_transform(testX)
testX = pd.DataFrame(testX, columns=colnames)

yhattest = model.predict(testX)
```

```
confM = pd.crosstab(yhattest, testY)
confM
```

extended	0	1
row_0		
0	27916	173
1	754	1028

```
accuracy_score(yhattest, testBrnli)
```

```
0.968966556191624
```

```
#0.968966556191624 extended
#0.7991362860299287 success
```

```
#classification trees
#0.969905935125364 extended
#0.865195996384695 success
```

```
#Creating a dataframe that implements accuracy for both methods. Classification tree done through another notebook
#that was using R instead.

data = [["Full Tree", 0.865195996384695, 0.969905935125364], ["Naive Bayes", 0.7991362860299287, 0.968966556191624]]
df = pd.DataFrame(data, columns=["Method", "Success Accuracy", "Extended Accuracy"])
df
```

	Method	Success Accuracy	Extended Accuracy
0	Full Tree	0.865196	0.969906
1	Naive Bayes	0.799136	0.968967

Classification Tree:

```
library(tree)
```

```
library(dplyr)
```

```
df = read.csv('C:/Users/andre/Downloads/clean_gtd.xlsx - Sheet1.csv')
head(df)
```

X	iyear	imonth	extended	latitude	longitude	vicinity	crit1	crit2	crit3	...	Biological	Chemical	Explosives	Fake.Weapons	Firearms	Incendiary	Mele
67505	1998	1	0	55.75138	37.579914	0	1	1	1	...	0	0	1	0	0	0	
67506	1998	1	0	54.60771	-5.956210	0	1	1	1	...	0	0	0	0	1	0	
67508	1998	1	0	31.99597	35.271110	0	1	1	1	...	0	0	0	0	1	0	
67522	1998	1	0	43.28036	-2.171588	0	1	1	1	...	0	0	1	0	0	0	
67523	1998	1	0	28.58584	77.153336	0	1	1	1	...	0	0	1	0	0	0	
67535	1998	1	0	36.80000	4.266667	0	1	1	1	...	0	0	1	0	0	0	

```
df2 = select(df, -c(1,2,3,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26))
head(df2)
```

extended	Australasia...Oceania	Central.America...Caribbean	Central.Asia	East.Asia	Eastern.Europe	Middle.East...North.Africa	North.America	South.America	So
0	0	0	0	0	1	0	0	0	
0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	1	0	0	

```
df = select(df, -c(1,2,3,5,6,7,8,9,10,11,12,14,15,16,17,18,20,21,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,47,48,49,50,51,
head(df)
```

extended	success	nkill	nwound	Armed.Assault	Assassination	Bombing.Explosion	Facility.Infrastructure.Attack	Hijacking	Hostage.Taking..Barricade.Incident.
0	1	0	3	0	0	1	0	0	0
0	1	1	0	1	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0
0	1	0	44	0	0	1	0	0	0
0	1	0	0	0	0	1	0	0	0

```
df["success"][df["success"] == 1] = "Yes"
df["success"][df["success"] == 0] = "No"
df$success = as.factor(df$success)
head(df)
```

extended	success	nkill	nwound	Armed.Assault	Assassination	Bombing.Explosion	Facility.Infrastructure.Attack	Hijacking	Hostage.Taking..Barricade.Incident.
0	Yes	0	3	0	0	1	0	0	0
0	Yes	1	0	1	0	0	0	0	0
0	No	0	1	1	0	0	0	0	0
0	Yes	1	0	0	0	1	0	0	0
0	Yes	0	44	0	0	1	0	0	0
0	Yes	0	0	0	0	1	0	0	0

```
set.seed(5151)

training = sample(1:nrow(df), size = .70*nrow(df), replace = FALSE)
ttree = tree(success~., data = df, subset = training) #full tree
summary(ttree)
```

```
yhat = predict(ttree, newdata = df[-training,]) #Full tree
ytest = df[-training,'success']
ypred = ifelse(test=(yhat[,2]>0.01),'Yes','No')

fullTable = table(ypred,ytest)
fullTable
```

```
     ytest
ypred    No   Yes
  No    669     1
  Yes  4026 25177
```

```
accuracy1 = ((fullTable[1,1] + fullTable[2,2]) / (fullTable[1,1] + fullTable[1,2] + fullTable[2,1] + fullTable[2,2]))
accuracy1
```

0.865195996384695

```
df2["extended"][df2["extended"] == 1] = "Yes"
df2["extended"][df2["extended"] == 0] = "No"
df2$extended = as.factor(df2$extended)
head(df2)
```

extended	Australasia...Oceania	Central.America...Caribbean	Central.Asia	East.Asia	Eastern.Europe	Middle.East...North.Africa	North.America	South.America	So
No	0	0	0	0	1	0	0	0	
No	0	0	0	0	0	0	0	0	
No	0	0	0	0	0	1	0	0	
No	0	0	0	0	0	0	0	0	
No	0	0	0	0	0	0	0	0	
No	0	0	0	0	0	1	0	0	

```
set.seed(3999)

training = sample(1:nrow(df2), size = .70*nrow(df2), replace = FALSE)
ttree = tree(extended~., data = df2, subset = training) #full tree
summary(ttree)
```

```
Classification tree:
tree(formula = extended ~ ., data = df2, subset = training)
Variables actually used in tree construction:
[1] "Hostage.Taking..Kidnapping." "Bombing.Explosion"
Number of terminal nodes:  3
Residual mean deviance:  0.135 = 9406 / 69700
Misclassification error rate: 0.02922 = 2037 / 69701
```

```
yhat = predict(ttree, newdata = df2[-training,]) #Full tree
ytest = df2[-training,'extended']
ypred = ifelse(test=(yhat[,2]>0.01),'Yes','No')

fullTable = table(ypred,ytest)
fullTable
```

```
     ytest
ypred    No   Yes
  No  27892   147
  Yes   752  1082
```

```
accuracy2 = ((fullTable[1,1] + fullTable[2,2]) / (fullTable[1,1] + fullTable[1,2] + fullTable[2,1] + fullTable[2,2]))
accuracy2
```

0.969905935125364