A Comparative Study of Machine Learning Algorithms in Predicting Terrorist Incidents

**Harun Dahir Abdulqadir**

A thesis submitted for the degree of

Applied Data Science

Department/School of Mathematics

University of Essex

Date of submission for examination (November 2021)

Section I

Introduction

**1.1 Motivation**

The meaning of the term terrorism has been widely disputed by various political analysts. The narrative found in common media conveys terrorists as ‘crazy extremists who commit indiscriminate acts of violence, without any larger goal beyond revenge or a desire to produce fear in an enemy population’. This misinterprets terrorists as nihilistic actors who only seek lethality and operate outside the realm of rationality. As difficult as it is to believe, terrorism is rational and can rightly be understood as ‘a set of tactics employed by rational actors within a wider strategy of coercive political communication’ (E-International Relations, 2016). Much of research and scholarship on terrorism is centred around political organisation and psychological perspectives of terrorism, however this paper will aim to delve into the machine learning algorithms and data mining models as it pertains to analysing features in large datasets. The application of data science is something that is permeating through the world as we speak and as to are terrorist networks. The manifestation of a scientific discipline and the emergence of greater political threat creates a relationship worth investigating and developing further. Machine Learning algorithms and Artificial intelligence are embedded in various other fields such as healthcare, retail, business and even sports. Political science has also developed but I believe, not the extent that it can.

**1.2 Contributions**

This paper aims to tackle the issues surrounding global terrorism by implementing machine learning algorithms that would be able to predict the ‘successes’ of a terrorist attack happening in the future, but only as far as the observed algorithms will allow. Three Machine Learning algorithms will be analysed and evaluated for their performance in effectively predicting the probability of terrorist attacks in the future. The dataset has been gathered from the GTD, the most comprehensive study of terrorist incidents in the world.

**1.3 Relevant Literature**

As previously mentioned, the theoretical doctrines that underpin terrorism have widely been studied across numerous disciplines in a bid to understand the causes and how to develop operative counterterrorism machinery to pre-empt the threat of a terrorist. Much of this lies in the capabilities of data analysis and the various machine learning algorithms, whether it is in the scope of tracking the geographical dimensions of international terrorism with GIS (geographic information systems) or whether it is predicting the success of particular functions in forbearing attacks. The literature surrounding the topic is surprisingly rich and immensely fascinating, evaluating it will prove important in filling in gaps in our work while also finding light in the research of others.

The significant need to develop technology for counter-terrorism apparatuses has long been desired. One such technology was developed for these exact purposes and measured against the Olympics in Athens, 2004 (Singh et al., 2004). Their adaptive safety analysis and monitoring (ASAM) system which can underline potential terrorist threats in real time combines the hidden Markov models (HMM) and Bayesian networks (BN) to forego real time situations with hypothetical data (such as the Olympics) punched into their modelling process. Importantly, they note the importance of tracking antecedent terrorist activities and other precursors in thwarting the potential realisation of another attack. that enable attacks to go forth. Such a research project is largely comparable to that of a pair of researchers for homeland security.

The research project included a risk model which was developed to ‘calculate the terrorism risk level of different locations’ (Toure and Gangopadhyay, 2016). Naturally, this required incidents data in which could be measured against a number of set rules to predict future terrorist activities. Their risk model was ultimately based on ‘frequency and time factors’, presumably minimal in an effort to specifically predict the occurrence of an attack. After applying their weighted risk model to Baltimore shooting data which importantly carried longitude and latitude information, their results recalled that the model can predict ‘within a 1.5 miles radius incident that will occur in the next 24 hours’, providing ‘high precision values of up to 96.30%’ (Toure and Gangopadhyay, 2016). Ultimately an effective predictive risk model that demonstrably addresses a succinct aspect of terrorism that is without a doubt valuable in improving counter-terrorism measures. Despite this, the lack of an alternative data variables limits them in tracking the probability of occurrence rather than a more detailed analysis of specific activities. Saha et al in a paper from 2017 manage to do this very well by employing various different variables (Saha, Aladi, Kurian and Basu, 2017).

Saha et al manage to utilise the widely sourced GTD open database in a more extensive fashion. The paper aims to predict the various differing tenets involved in a terrorist attack. The analysis focuses on predicting the attack type, used weapon type and target type by using a plethora of learning algorithms (Saha, Aladi, Kurian and Basu, 2017). Saha et al are, however, not the only researchers to carry out such extensive analyses on the GTD dataset. Agarwal et al, provide detailed analysis on ‘the factors that might give blow to an increase of terrorism’ by aiming to predict the number of people killed (nkill), success of an attack, which group carried out the attack and interestingly, the effect of external factors such as weather on a terrorist attack (Agarwal, Sharma and Chandra, 2019). A comparison of data mining techniques such as logistic regression, support vector machines and random forest to attain the best results were used with accuracies of up to 95%. Similarly, Mo et al, carried out almost identical analyses to predict future terrorist attacks with the same dataset, although this time employing SVM, Naïve Bayes and log regression to obtain precision of 78% (Mo, Meng, Li and Zhao, 2017).

Along similar lines, Talreja et al continued the work on the GTD dataset and curated a predictive model that could dial in on the terrorists/terrorist groups carrying out the attacks. The variables they designated through their feature selection ‘Factor Analysis of Mixed Data (FAMD)’ were then button into a number of ML algorithms, although in this instance, the trained model failed to correctly predict which perpetrating groups, producing inaccurate results.

Another utilisation of machine learning algorithms was employed by Uddin et al. in which they develop five different models based on deep neural networks. ‘The performance of the DNN is compared with NN and the three machine learning algorithms, and it is demonstrated that the performance in DNN is more than 95% in terms of accuracy, precision, recall, and F1-Score’ (Uddin et al., 2020). Their results were impressively evaluated and contrasted next to other ML to prove efficacy and find the most accurate results possible.

In essence, it seems all of the previous research papers in the literature have applied either machine learning algorithms or artificial intelligence deep learning models to analyse or predict terrorist attacks. Most of the novel and leading papers follow the notion of highlighting patterns within previous terrorist attack and suggest alternative solutions to help thwart them. Only a few of the papers we observed are advanced in predicting future terrorist attacks and if they are, there has simply not been enough research done, particularly along the lines of success, region, and suicide. As such, it would be fair to suggest that there is a research gap in the literature for applying various machine learning algorithms to predict a future attack.

Section II

Exploratory Analysis

The widely used dataset is sourced from the GTD, previously owned, and now supported by the National Consortium for the Study of Terrorism and Responses to Terrorism (START). The dataset is a database of terrorism attacks spanning from the 1970s all the way to 2019. The complexity and the well drafted nature of the database is arguably its biggest strength with 135 variables and over 200,000 observations. Hailed as ‘currently the most comprehensive unclassified data based on terrorist events in the world’ (Mo, Meng, Li and Zhao, 2017). The variety in the database offers researchers and analysts the luxury and the freedom to provide a host of different analyses on different issues in the literature, as observed in the number of scholarly papers the GTD dataset has featured in.

In lieu of applying the methodology for our study, I believe it will be important to indulge in exploratory analysis of the dataset, observe which variables are of particular interest to us and are the most correlating. As mentioned, the database offers a wide variety of variables to carry out our exploratory analysis, however we will aim to analyse:

* **Killed**: The killed variable denotes the number of people killed during the attack. This includes the number of perpetrators.
* **Geographical Features**: Multiple variables such as longitude, latitude, region, country pinpointing where the attacks took place.
* **Success**: Success is a categorical variable with two values. It takes the value 1 when the attack is a success and 0 otherwise.
* **Suicide**: Suicide is also a categorical variable with two values. 1 when suicide and otherwise 0.

Several columns were especially interesting to me from a text analysis perspective which allowed for a more expansive use of exploratory tools.

* **Summary**: Summary provides a somewhat detailed description of the incident.
* **Motive**: Motive highlights the motivation behind an attack.
* **Target Type:** Classification of the individual/group targeted by attack.
* **Attack Type**: General method of attack and broad class of tactics used.

Underlining these variables will allow us to better understand terrorist activities throughout history and simply enable us to draw some early insights into the intricacies of the data available to us.

**Terrorism Across Countries by Year**

Map

Description automatically generated

Figure

After plotting the points on the world map, we can vaguely see the regions that are most affected and least affected. For example, regions such as MENA (Middle East and North Africa), Africa, and South Asia have been heavily affected by terrorist attacks over the years. Regions such as North America, as it pertains to Canada have been fortunate to not have so many terrorists’ attack. In comparison to their neighbor the USA, it is littered with terrorist attacks, most of them being domestic as we will later observe.

Understanding the total amount of attacks that have affected each country is important insight but what is more important is highlighting the change in attacks per country over time. From separate analysis, we observed that in the North America in the 1970s suffered the most and in the years 1970-2000, Laite et al note that countries such as ‘Colombia, India, and Great Britain suffered the most from terrorist attacks. Nonetheless, when analysing the years 2001-2017, there is a major shift in terrorism operations from America and Europe to the Middle East and Southeast Asia (Laite, Lozano and Sankaranarayanan, 2019). Sub-Saharan Africa also became a much more affected region in that later period. It is extremely insightful to underline this apparent evolution in the where the attacks were frequently located over time.

North America and Europe became something of a safe haven. This can be attributed to what can be seen as a potential drawback of the GTD dataset. Given that the GTD is built and collated in the US and the western world, it is perhaps biased as a product of global development. Many of the attacks reported in the early years were less likely to be reported in underdeveloped countries and regions, which is perhaps why the 1970s saw the USA as the most affected country.

Another intriguing country is the Philippines, with over 1000 terrorist attacks and 313 coming since 2015. Interestingly, the Philippines has been heavily affected for a long period of time, perhaps attributed to their long history of religious sectarianism which can be observed in the motive feature of the dataset.

**Deaths by Region**

A more detailed approach to terror by region is demonstrated here by observing the amount of people killed by these attacks over the years. As observed in the world map however, we can confirm that MENA and South Asia suffer the most deaths over time, with MENA reaching heights of 6518 deaths in 2007 and 5401 deaths in 2014.

**Chart, histogram

Description automatically generated**With some context, MENA and South Asia have had higher death rates as much larger regions with more countries packed in, as opposed to somewhere like North America. An understanding of the political climate in these regions helps us understand further the causes for these spikes in deaths/terrorist attacks, for example the Arab Spring in 2014.

Figure

**Frequency of Terror Attacks by Attack Type**

**Chart

Description automatically generated**

Figure

Here we observe that ‘Bombing/Explosion’ has always been the most frequent type of attack used by perpetrators aside from one year in 2011, aside from this ‘Armed Assault’ has always closely followed. Often time these two overlapped in attacks which would explain their similar patterns and trajectories.

Chart, histogram

Description automatically generatedBelow we highlight the same data but in a line plot, a more readable format.

We can take a closer look here of the same data but in a different format to close understand the small discrepancies and how Armed Assault was in fact more frequent for two years in 2010 and 2011, observing the highest number of attacks by any one attack type in a calendar year (1063).

**Success Rate of Terrorist Attacks**

**Chart, histogram

Description automatically generated**

The success rate of terrorist attacks is a particularly interesting variable to look at as we can see when both the most successful and unsuccessful periods were and how they coincide. The lighter line closer to 1.00 is evidently represented by the success of the attacks and the adjacent line is represented by the failure rate. The graph highlights the minimum and maximum points of success/failure depending on which perspective you choose to interpret from. Multiple line graphs like these are effective in showing trends over time and in this one, the success rate of terrorist attacks have been particularly steady over time. The most peaceful period regarding a lack of terrorist attacks has been in recent times, following the year 2010 it can be seen there has been a steady drop in terrorist attacks. The highest failure rate was observed and in 2017 and the highest success rate was observed between the years 2005-07

Text

Description automatically generated**Motive (Text Analysis)**

In providing some text analysis on the motive variable we subsampled around 1% of entries to be randomly sampled. This text analysis is useful in helping us create a visual representation of word frequency, the more frequently the words appear in the text, the larger they will be in the word cloud.

Applying the ‘stop words’ in this scenario has understandably still thrown out subject specific words such as ‘motive, attack, specific, sources, incident, targeted’. However, we will look beyond that and have a look at the slightly smaller words such as ‘Sunni’, ‘Shiite’, ‘Maoists’, ‘Sectarian’, ‘Destabilize, and ‘Responsibility’. From these words alone we can begin to piece together an understanding of where most of the motives come from. Most of them are seemingly religious and political, furthering our understanding that terrorism is almost always politically driven and never a random, nihilistic act of violence.

**Summary**

Exploratory analysis is almost always simply to gauge an understanding of a dataset before further providing analysis on a specific subject, and as such, this exercise has only scratched the surface. Our brief overview of the most affected regions highlighted to us those countries such as Australia, Russia and Canada suffer the least from terrorist attacks, while countries such as Iraq, India, Afghanistan, Pakistan, and Colombia have suffered the most. It is also evident that ‘Bombing/Explosion’ was the most frequent attack type employed by terrorists in their attacks and ‘Armed Assault’ followed closely behind. Predictively, ‘Bombs/Explosions’ are frequently the most used weapon type. We saw that in recent years there has been an increase in the number of terrorist attacks, but the success rate has not been steadily increasing in line with it. This could be attributed to the improved efforts of government counter-terrorism projects and the collective work of anti-terrorist organizations.

Section III

Method and Methodology

As opposed to common analyses and in contradistinction to seminal classification projects in which ideological typologies are scrutinised for adding subjectivity to their findings, this study has found the profound advantages of operationalising various machine learning (ML) techniques to gain a better understanding of the distribution of current terrorist attacks and the trends that make them succeed. The successes of Machine Learning are not difficult to notice in almost every profession with ‘AI being leveraged to…formulate policy, guided targeted advertising, detect and fight disinformation, and…combat the spread of disease’ (Schwartz, 2021). Not far removed from these instances, we can also underline its efficiency in affecting business models, biomedicine, and fintech and energy. The various ways machine learning and artificial intelligence have been optimised behind the scenes to impact everyday lives and operations was evidence enough to demonstrate to us how we can we find patterns in this dataset that will be able to help drive decisions and impact future terrorist attacks.

In the same breath, we thus decided to utilise the machine learning algorithms Logistic Regression (LR), Support Vector Machine (SVM), and Naïve Bayes (NB) to help us measure how well we can predict the success of terrorist attacks in the GTD dataset and then run a comparison of the implementation of these classification models. Before we arrive at the analysis of the classification models, we clean the dataset through pre-processing and then apply two feature selection models: Boruta and Backwards Regression (using p-values) to select optimum features for the classification models.

In this section we will be annotating a detailed analysis of the proposed dataset. This paper will aim to break down the methodology into four parts:

* Overview of the Dataset
* Data Pre-Processing
* Classification Models

**1.1. Data Set**

As we know and have previously lamented, the famous GTD data set endorsed and supported by the National Consortium for the Study of Terrorism and Responses to Terrorism (START), is a detailed study of every event that concerns terrorism ranging from 1970 to present day. With a high degree of impunity and investment in homeland security, the GTD is the most comprehensive study of terrorist incidents in the world. While the GTD has few setbacks in providing such a vast amount of information on terrorism, about 30% of the dataset has ‘unknown’ values such as not knowing which terror group carried out which attack. To be a truly thorough and complete catalogue that can be developed for modelling predictive analysis, putting a name to terrorist attacks are vital. However, as we know, no dataset is perfect and missing or incomplete values are something that needs to be managed (Mo, Meng, Li and Zhao, 2017).

**1.2. Pre-Processing**

As we know the GTD includes a vast amount of information, spanning well over 100,000 cases of almost all terrorist events since 1970. To identify each of these instances, the variables corresponding to the date and location of the attacks are the most important, along with ‘eventid’ to help us distinguish between each incident. Overall, there are 132 attributes that do extremely well in detailing almost every aspect of the recorded terrorist attacks, varying from geographical locations to motives. The GTD mention that the data was collated from a number of various different sources which suggests that it may result in data inconsistencies. To avoid any particular problem like this we used roughly 10% for the explanatory data analysis to draw sound inferences from attributes that share complete data from across the board. Similarly, we aimed for a similar outcome when modelling our machine learning algorithms. The threshold for the share ratio was set for 30%, signifying that only those variables which are included in more than 30% of the data will be included, which is something of an early pre-processing operation. After considerable pre-processing and narrowing down our data through feature selection, which we will discuss, we were left with 45% of operational data.

Although operational and further focused, we still had the issue of missing values to deal with. To solve this problem, we applied data imputation with the MICE package in R, which is often more commonly known as Mean/Mode/Median Imputation (MI). We will dive further into this later in the chapter.

**1.3.1 Feature Selection**

Feature selection in many ways is integral in developing a predictive model. Through the many methods finely curated and developed over time, it aims to reconstruct a version of particularly large and challenging datasets that are still optically desirable and easier to grasp. Of course, the main feature of the feature selection is refining the number of input variables that carry out your predictive model, in turn reducing the computational cost that accompanies large datasets and ML, and also usually improving the performance of any said algorithm/model.

It can often be difficult for a statistician or data scientist to find the right feature selection method to match their dataset, and as such, a refined understanding of what feature selection methods involve can be helpful. Brownlee underlines that ‘statistical-based feature selection methods involve evaluating the relationship between each input variable and the target variable using statistics and selecting those input variables that have the strongest relationship with the target variable’ (Brownlee, 2019). The wide variety of methods can often be characterised into three main types of selection methods:

* Filter Methods: the selection of features is independent of any machine learning algorithm, and instead the features are based on scores of a host of statistical tests and their correlation to the outcome variable. Statistical tests include some common correlation measures such as Pearson, Fisher’s Score, Chi-Squared test and Anova.
* Embedded Methods: a set of algorithms that have their pre-set built in feature selection methods such as that of LASSO regression.
* Wrapper Methods*:* wrapper methods tend to involve training a full model with a subset of features of and the resulting inferences aid you in keeping or removing features from the previous subset of features. Some examples include Forward Selection, Stepwise, Backward Elimination and Boruta, which we will be utilising for our own feature selection process.

**1.3.2 Boruta Algorithm**

As previously mentioned, the feature selection umbrella that the Boruta method falls under is simply to help us gain greater precision in our models’ performance by reducing the noise from our data. Boruta is one of the wrapper methods that is centred around the random forest classification algorithm. With the afterthought of the dependent variable, the algorithm aims to zero in on the most significant variables (Sharma, 2021).

It begins by creating shadow features in its own duplicated dataset. It then trains a classifier, most often Random Forest Classifier. By carrying this out on the shadow features in the shadow dataset it allows for an idea of the level of importance of each feature judging by their scores. After this, the actual dataset is checked for its features in comparison to the Z-scores of the shadow features. If they do, they are recorded as hits and nominated in a vector where they are constantly compared to further iterations. At every iteration the random classifier continuously compares the shuffled copies to the original features and if it that is in fact the case then it is marked as an important feature.

The image below demonstrates the method.

Graphical user interface, application, table

Description automatically generated

The natural assumption when applying the random forests classifier sourced method is that the more attributes you have the better. However, removing insignificant variables rather contributes to achieving higher model accuracy. Focusing on understanding the theory it is not the be all end all, so Chart, waterfall chart

Description automatically generatedwe demonstrate how it has applied to our dataset.

The plot shows us that all 8 of our variables are deemed important to our model in predicting the success of a terrorist attack and none are considered ‘unimportant’. It also highlights to us that the minimum, maximum and average shadow scores. When the variables are highlighted in green it lets us know that the predictor is important, alternatively when it colours red it lets us know that the variable is rejected and deemed unimportant. A yellow box suggests that the variable is tentative, which refers to an importance score that is so close to the shadow attribute that Boruta has been unable to decide whether it is significant or not in its iterations. Essentially highlighting that its undecided in its number of runs.

Below we confirm the assumption from the Boruta plot.

getConfirmedFormula(boruta1)

## success ~ year + region + suicide + attacktype + weapon + target +   
## killed + wounded  
## <environment: 0x00000000384939e8>

However, testing our model against just one method, despite how telling the results may be, is not assured so we will run our model against another method, backwards stepwise regression.

**1.3.3. Backwards Stepwise Regression**

Backwards stepwise regression is a stepwise regression approach from the same family that starts with a full model and all of the dependent (response) and independent (predictor) variables being inserted into the regression method. At each junction the variables that are not befitting are gradually removed from the regression and a reduced version is produced that best fits the model. It is termed backwards because it removes predictors to begin with and if the r-square value changes in the new model then this one is preferred…this process is named backwards elimination regression. It is vital in helping us protect against the multicollinearity problem and goes a long way in helping us avoid overfitting.

## Model Summary   
## --------------------------------------------------------------  
## R 0.248 RMSE 0.191   
## R-Squared 0.061 Coef. Var 19.943   
## Adj. R-Squared 0.061 MSE 0.037   
## Pred R-Squared 0.060 MAE 0.076   
## --------------------------------------------------------------

Application of this method to our model resulted in what was already confirmed with our previous model. The backwards regression aimed to eliminate any insignificant contributors to the model, however there were no predictors with no incremental predictive power.

BWDfit.p

[1] "No variables have been removed from the model."

Variables were removed based on p-value and in this case, as with most, it was 0.05 or 5%. As with the Boruta method, none of the variables satisfied this condition, so none were removed.

**1.4. Selected Features**

Going into our pre-processing we had already narrowed down the 135 primary attributes in the dataset to around 15 and further reduced it to 7 independent variables for the application to our feature selection method, Boruta. The following different factors are what we will be training our machine learning algorithms to learn.

* **Success**: Success is a categorical variable with two values. It takes the value 1 when the attack is a success and 0 when it is otherwise. This will be our dependent variable
* **Suicide**: This field indicates whether the attack resulted in suicide or not. 1 = “Yes” suggesting it was a suicide attack. 0 = “No” subsequently means there has been no signal of a terrorist attack. Dimensions of the dataset are 75% utilized for training data (62009) and 25% for the testing data (31005). Continued throughout the variables.
* **Killed**: The killed variable denotes the number of people killed during the attack. This includes the number of perpetrators.
* **Region:** This variable details the region of the attack.
  + North America
  + Central America and Caribbean
  + South America
  + East Asia
  + Southeast Asia
  + South Asia
  + Central Asia
  + Western Europe
  + Eastern Europe
  + The Middle East and North Africa
  + Sub-Saharan Africa
  + Australasia and Oceania
* **Target Type:** Classification of the individual/group targeted by attack.
* **Attack Type**: General method of attack and broad class of tactics used. 9 different categories are noted throughout the dataset. These are:
  + Assassination
  + Armed assault
  + Bombing/explosion
  + Hijacking
  + Hostage taking (barricade incident)
  + Hostage taking (kidnapping)
  + Facility/infrastructure attack
  + Unarmed assaults
  + Unknown
* **Weapon**: Weapon type highlights the weapon used in the attack. The variable observes 13 different categories to describe which weapon was used. The categories are described as:
  + Biological
  + Chemical
  + Radiological
  + Left as blank
  + Firearms
  + Explosives
  + Fake weapons
  + Incendiary
  + Melee
  + Vehicle (not to include vehicle-borne explosives, i.e., car or truck bombs)
  + Sabotage Equipment
  + Other
  + Unknown

As vast as it may be, the GTD dataset was extremely detailed to the point that the naked eye was able to narrow down which features would be of use in a machine learning model. Many of them were sub attributes, character strings, and mostly variables that had over 50% of missing values. For this reason, the feature selection process was easier than expected and was only confirmed through our processes.

**1.5. Missing Data**

Datasets often carry or contain missing values for various reasons. They are often distinguished as NaNs, blanks, or any other placeholders. Training a model with a dataset that carries a high number of missing values will often cause errors in running a model and if not then it risks providing skewed results. The most common approach, aside from ignoring them and having your software handle them, is imputation. Imputation is simply replacing the values with an estimate and then completing the data set as if the imputed values were actual observed values (Jakobsen et al, 2017).

Much like feature selection, there are a number of ways to implement data imputation. Common methods for single imputation include:

* Mean Imputation: the value is calculated as a mean of all observations from said variable.
* Multinomial Logistic Regression Imputation: the value is predicted by performing regression on other variables from the same observation - often the choice for categorical variables.
* Stochastic Regression Imputation: the value is predicted similarly to log regression imputation but adds a random residual value.
* Hot-Deck Imputation: the value is randomly selected from an observation that holds similar values with other variables
* Cold-Deck Imputation: the value is systematically selected from an observation following similar values in line with other variables, similar to hot-deck without the random selection.

(Grace-Martin, 2017), (Buuren and Groothuis-Oudshoorn, 2011).

To carry out the data imputation for our dataset, we utilised the Logistic Regression Imputation in the ‘mice’ package in R as opposed to more popular methods such as that of Mean Imputation (MI) and Stochastic Imputation, namely because MI offered an average number with multiple decimal places whereas we were searching for an integer that that fit into our continuous variable quite easily.

Some limitations from the regression imputation included ensuring it is computationally feasible, although this wasn’t an issue as our variable wasn’t a large categorical variable with countless categories and the number of missing values wasn’t high. Another limitation is underestimating the accompanies regression coefficients since the impute values are estimates themselves. They carry subsequent random error that is left unaccounted for because it is implanted as a fresh observation and thus ignores the p-values and standard errors that are too small. However, this is a minute drawback that is worth noticing but not lamenting over (Zhang, 2016).

Chart, histogram

Description automatically generatedBelow is a visualisation of the missing values we encountered. Fortunately, the values were limited to 6792 observations across one continuous variable.

After operationalising the ‘mice’ package we were able to impute the data into the missing values in the ‘wounded’ variable and successfully merge the imputed data to complete the dataset. Importantly, check for any other missing values after the procedure to make sure and it seemed to check out.

year region success suicide attacktype weapon target   
 0 0 0 0 0 0 0   
 killed wounded   
 0 0

As for normalisation, there is not a glaring need to normalise the values of my numeric variables to a common scale as they do not have different ranges, it is only required when they have different ranges.

**1.6. Classification**

Supervised learning has long been understood and greatly surveyed in many different scientific disciplines. Supervised learning when labelled values are operationalised to formulate a predicted model is a protocol that is the most dependable and used in data mining/science. A few of the most famously used and adopted supervised classification models include Support Vector Machines (SVM), Logistic Regression (LR), and Naïve Bayes (NB). Logistic regression is perhaps the most widely adopted for its ‘parametric method to analyse dichotomous response variables and find the relationship between the response variables and independent variables’ (Musa, 2012). It is characterised for its predictive fervour in healthcare and business models, yet for all of its in-depth statistical analysis, it is less known for its uses in Ai and ML. Support Vector Machines (SVM) algorithm is more broadly adopted in machine learning for its relative ease and understanding and has been applied to real life fields of work (Small and Roth, 2010). The Naïve Bayes classifier is from the Bayesian learning theorem and is based on the simple ‘assumption that the attributes are conditionally independent given target value’ (Islam, Wu, Ahmadi and SidAhmed, 2010). Performance for most datasets may always vary depending on the algorithm or system used. As such a comparative study of a few algorithms applied to our model is important.

**1.6.1 Support Vector Machines**

Drawn from the statistic learning theory, SVM adopts the structural risk minimisation principle. SVM is famous for, and widely adopted, for ‘solving the two-class recognition problem’ (Mo, Meng, Li and Zhao, 2017). SVMs possess the acclaimed ability of being ‘universal approximators of any multivariate function to any desired degree of accuracy…consequently, they are of particular interest of modelling the unknown, or partially known, highly nonlinear, complex systems, plants or processes’ (Wang, 2005). Essentially, the main function of the classifier is to use a hyperplane classifier to augment the fine margins and separate the positive and negative data features (Burges, 1998).

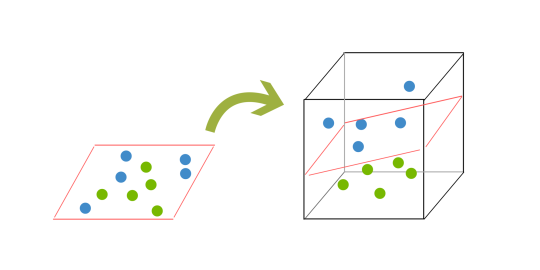
Below is a formulaic definition of how SVM operates.

A separating hyperplane can be defined as,

where is a weight vector and b is a bias.

In application of any given training data, although there are many alternative hyperplanes which may do the same thing and separate features between a hyperplane, the SVM classifier is based particularly on a hyperplane which ‘maximises the separating margin between the two classes’ (Mo, Meng, Li and Zhao, 2017). The assumption of the SVM is that a hyperplane which splits the features, but also is as far away from any training point as possible, will generalize best. The optimisation problem can be stated as this:

With C being the required condition parameter, the variable is a vector containing the slack variables , i = 1,2…n. Carrying the purpose of finding an optimal separating hyperplane (Mo et al, 2017).

Simply put the goal is to find the right hyperplane that linearly separates and classifies a set of data. Although ‘linearly separating’ helps us to grasp the concept better, the hyperplane is more like a three-dimensional plane that is mapped into higher and higher dimensions until it segregates the data. Finding the right hyperplane thus entails the greatest possible margin between any point in the training set and the hyperplane itself. When this is satisfied any new data being fed into the model has a much greater chance of being classified correctly.

**1.6.2 Logistic Regression**

Logistic regression is a classification model that essentially estimates the probability of an outcome based on the relationship to the independent variables, whether it has one or more predictor variables and a response variable. There are three types of logistic regression:

* Binary Logistic Regression: the target variable or the dependent variable is binary in nature and only has two possible categorical values (1,0)
* Multinomial Logistic Regression: the dependent variable in a multinomial carries three or more outcomes/values dependent on the independent variables
* Ordinal Logistic Regression: similar to multinomial logistic regression where the dependent variable has tree or more outcomes but in this case the outcomes are ordered.

One of the many uses and functions of logistic regression is that the response variable can be either categorical or continuous, unlike linear regression which requires strictly continuous data. Our response variable, however, will still be a binary value. Comparatively to linear regression, logistic regression also allows us to use log odds ratio rather than probabilities and an iterative maximum likelihood method in place of a least squares method to fit the output. Logistic regression can be expressed formulaically as such:

Where the beginning of the equation represents the logit or the ‘log-odds’ function, and the equation within the brackets p(x)/(1-p(X)) represents the odds. The odds as you might imagine is the calculated ration of the probability of success to probability of failure. If we process the equation in an inverted format it shows:

This here is understood as the Sigmoid function as it always delivers a value of probability ranging from 0<p<1. When plotted, the sigmoid function always gives an S-shaped curve.

**1.6.3. Naïve Bayes**

The Bayesian learning algorithm, as previously mentioned, takes an extremely practical approach for most learning problems. Bayes himself theorised that the probability of future events could and should be determined by evaluating their earlier frequency. This discrete simplicity is what allows it to often outperform some other more complex classification algorithms. It functions relatively quickly with large datasets and is most commonly used for text classification (Islam et al, 2010). It indiscriminately deals with any number of attributes or categories and performs surprisingly well. Naïve Bayes’ functionality takes from the Bayes theorem which states that:

P(Y|x) is the posterior probability, P(Y) is the prior probability of class, P(x|Y) is the likelihood, and P(x) is the prior probability of predictor. Leading on from this, employing the naïve independence assumption looks like:

The Naïve Bayes classifier is simple in that it is based on the simplifying assumption that the attribute values are conditionally independent given the target value (Mitchell, 2017). Essentially, the belief is that the target value is simply the amalgamated product of all the probabilities of their respective variables. To find the most maximally probable hypothesis, MAP is utilised in this method which is known as the Maximum A Posteriori decision rule in a Bayesian setting. (Mo et al, 2017). We can use the following classification rule:

As we said the Maximum A Posteriori estimation is utilised to the estimate the P(Y) and . The uncomplicated fashion in which the Naïve Bayes model estimates the probability of variables is beauty of the model. It works in way that assumes that the presence of a particular feature in a class is unrelated to the presence of any other in the dataset.

**1.7. Justification**

A main feature in meeting the targets of efficiently testing our research question was founded in constructing, deploying, and operationalising the right classification methods for our model. As previously mentioned, the three machine learning algorithms logistic regression, support vector machines and naïve bayes all carry their respective strengths and are utilised in a number of different domains, each one being appropriate for the ML method chosen. Some datasets and the models that are curated from them perform well as a response to particular methods but be perform inadequately on others. Even more succinctly the performance metrics associated with them, for example a particularly large dataset may not function as superbly with the various regression coefficients in a Maximum Likelihood Estimation as it might the Maximum A Posteriori decision rule in the Bayesian setting. Subsequently, it was imperative to compare the predicting capabilities of these three methods on our dataset.

Naïve Bayes was selected due to its proficiency in real time predicting. It is commonly renowned for its fast computational times and is an eager learning classifier. Although it is often used for text classification and sentimental analysis, it is also useful for multi-class prediction features, predicting the probability of multiple classes of response variable. Comparatively, logistic regression is a lot more detailed with low variance. Although not as fast or adept at handling large numbers of categorical variables, it is simple and efficient and provides probability score for observations as a function of the corresponding output.

Much like logistic regression, SVMs can also be hailed for adopting categorical response features that can operate with three or more values. On the other hand, the kernel process in SVMs can also be employed in logistic regression – ‘kernel logistic regression’. While logistic regression aims to make good use of all the data points, ‘points relatively far away from the margin have less influence because of the logit transform’ (Nadkarni, 2016). Therefore, as different as the processes may be logistic regression and SVM end up giving the somewhat similar results. For this reason, I felt it was important to measure the two machine learning models against one another and try both. As with naïve bayes, SVMs can sometimes be a better fit and perhaps more computationally efficient, whereas logistic regression will often always be more valuable in the interpretability.

Section IV

Results

In our experimentation with this dataset, to comprehensively investigate the performance capabilities of our classifiers, it is important that we split the training set into training and testing sets to the ratio 3:1. The training set is randomly selected and split with functions from the ‘caTools’ package in R. To evaluate the performance results correctly and effectively by precision, we will be observing the mean value of all results and the standard error. The aim of our study in applying this GTD dataset was to see how well we could predict the success of a terrorist in the future. Estimating the performances will be a product of our pre-processing and the subsequent subset features that we have selected and explained in various steps previously.

The format of these results will take the shape of gauging each model and its predictive process and then further evaluating their computational time. Following this, the paper hopes to carry out a comparison of the classification models and their precision.

**Naïve Bayes**

As previously mentioned, we split the data into training and testing data at a 3:1 ratio.

Calendar

Description automatically generated

Figure 7

When multicollinearity occurs and independent variables are correlated, it indicated that in one variable are associated with shifts in another variable. And as such, the stronger the correlation the harder it is to change one variable without changing the other. The naïve bayes classification model would be hindered by this so the first thing to do , was to make sure that the independent variables are not highly correlated. As we can see, the correlation coefficient in the figure 7 is actually only 0.38 so there is not such a high level of multicollinearity.

## p1 0 1  
## 0 863 5247  
## 1 1585 54314

Just above we can see the confusion matrix from the naïve bayes model. In this we can infer those 863 times the model correctly predicted that the terrorist incident was not a success and 54314 times it correctly predicted that the terrorist attack was a success.

However, we can see that there are quite a few misclassifications from our naïve bayes method. We employed the classification error formula which looks like this:

**(FP+FN)/(TP+TN+FP+FN)**

 or

(1-Accuracy)

This formula helps us calculate what fraction of predictions were incorrect in our model and just below we can see that roughly 11% of our predictive model was misclassified.

## [1] 0.1101776

**Logistic Regression**

**Chart

Description automatically generated**Comparably to our multicollinearity issue in our naïve bayes model, the log regression deemed the variables ‘region’, ‘weapon’ ‘killed’ all statistically insignificant and as such we chose to remove those features from the model, leaving us remaining with ‘wounded’, ‘attack type’, ‘year’, and ‘suicide.

Through calculating the previous calculation and measuring the predictor values, it is possible to find out the probability of success for any instance in the dataset, particularly if you have the standard deviation and mean. What we can gather from the table is that the logistic regression, as a process, meaning providing the statistics to help me eliminate the insignificant variables, have seemingly helped the accuracy of the model exponentially. We will have a look at the confusion matrix now and compare.

## Actual  
## Predicted 0 1  
## 0 87 88  
## 1 2346 59489

Following the method from the previous ML algorithm, we apply the same classification error formula here and find that the error is actually significantly smaller than that of naïve bayes method.

## [1] 0.03926786

**Support Vector Machines**

Although we boasted the well-known benefits of implementing a SVM method that could help us run a bulkier comparison of machine learning algorithms, we actually could not manage the computational cost as we seemingly did not have the resources. My apparatus is an “XPS 13 7390” with an 8.00GB Installed RAM. It was seemingly not enough as computational time was exceedingly over almost 8 hours. With 93,104 total observations, it could have been that the iterations were simply too large in the search for the right hyperplane. Our methodology did however highlight this and mentioned that the algorithm is not suitable to large datasets.

Nonetheless, we have thoroughly studied that the theoretical groundings of SVM and have a greater understanding for the algorithm. In future, it will be perhaps wiser to seek a stronger computer or simply allocate more time to be able to run the algorithm.

SECTION V

Summary and Discussion

I believe we satisfied our belief that Machine Learning could be used for mapping terrorism with both efficiency and accuracy. We observed how a well selected feature selection method can make all the difference and save you a heap of time and even still found that running particular models, such as that of logistic regression, would allow us to reduce noise and remove insignificant variables even at the very end. We concluded in the end that Logistic Regression outperformed the other ML method, Naïve Bayes which indicates to us that most of the relationships in the GTD dataset are most likely linear. Another point of interest is our failure to get the SVM method to operate functionally – although through no fault of our own. In the future, it might be better to test the algorithms first and what would be required to meet the conditions of their method.

Our machine learning methods were an impressive tool, despite the fact that they did not further narrow our variables. This can be attributed to the fact that they were already reduced from observing the dataset and have prior knowledge. A concern, however, may be that our logistic regression model demonstrated to us that there were insignificant variables that would hinder our accuracy, the exact function of the feature selection method. In summary, we can conclude that Machine Learning can go a long way in facilitating efficiency and precision for the data scientists and policy makers that work on counter terrorism. Logistic regression showed the finest margin for error, yet Naïve Bayes also did not have large classification errors. Nevertheless, it is an immensely compelling and promising research area that deserves more time and exploration. My future work would be to look into geospatial analysis and working with GIS and Big Data to predict attacks in real time.

**Bibliography:**

* Agarwal, P., Sharma, M. and Chandra, S., 2019. Comparison of Machine Learning Approaches in the Prediction of Terrorist Attacks. Twelfth International Conference on Contemporary Computing (IC3), pp. 1-7.
* Brownlee, J., 2019. *How to Choose a Feature Selection Method for Machine Learning*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/>
* Burges, C., 1998. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(121–167 (1998).
* Buuren, S. and Groothuis-Oudshoorn, K., 2011. mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, 45(3).
* Islam, M., Wu, Q., Ahmadi, M. and SidAhmed, M., 2010. Investigating the Performance of Naive- Bayes Classifiers and K- Nearest Neighbor Classifiers. *Journal of Convergence Information Technology*, 5(2), pp.133-137.
* Jakobsen, J., Gluud, C., Wetterslev, J. and Winkel, P., 2017. When and how should multiple imputation be used for handling missing data in randomised clinical trials – a practical guide with flowcharts. *BMC Medical Research Methodology*, 17(1).
* Laite, R., Lozano, M. and Sankaranarayanan, K., 2019. Terrorist Group Classification of Historic Terrorist Attacks from The Global Terrorism Database. 14th International Conference on Computer Science & Education (ICCSE), [online] pp. 237-242. Available at: <https://ieeexplore.ieee.org/document/8845417?figureId=fig1#fig1>
* Medina, R. and Hepner, G., 2011. Advancing the Understanding of Sociospatial Dependencies in Terrorist Networks. Transactions in GIS, 15(5), pp.577-597.
* Mitchell, T., 2017. *Machine learning*. New York: McGraw Hill.
* Mo, H., Meng, X., LI, J. and Zhao, S., 2017. Terrorist Event Prediction Based on Revealing Data. IEEE 2nd International Conference on Big Data Analysis (ICBDA) pp.239-244.
* Musa, A., 2012. Comparative study on classification performance between support vector machine and logistic regression. *International Journal of Machine Learning and Cybernetics*, 4(1), pp.13-24.
* Nadkarni, P., 2016. Core Technologies: Machine Learning and Natural Language Processing. *Clinical Research Computing*, pp.1-26.
* Schwartz, H., 2021. *The Evolving Role of Artificial Intelligence and Machine Learning in US Politics*. [online] Csis.org. Available at: <https://www.csis.org/blogs/technology-policy-blog/evolving-role-artificial-intelligence-and-machine-learning-us-politics>
* Shah, S., Aladi, H., Kurian, A. and Basu, A., 2017. Future Terrorist Attack Prediction Using Machine Learning Technique. Working Paper, pp 1-7.
* Sharma, S., 2021. *Boruta Algorithm | What is Boruta Algorithm | Boruta Algorithm in R*. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2021/05/boruta-with-r-is-a-great-way-to-address-the-curse-of-dimensionality/>.
* Singh, S., Haiying Tu, Allanach, J., Areta, J., Willett, P. and Pattipati, K., 2004. Stochastic Modelling of a Terrorist Event via the ASAM System. IEEE Potentials, 23(3), pp. 18-21.
* Small, K. and Roth, D., 2010. Margin-based active learning for structured predictions. International Journal of Machine Learning and Cybernetics, 1(1-4), pp.3-25.
* Start.umd.edu.2015. GTD | Global Terrorism Database [online] Available at, http://www.start.umd.edu/gtd> [Accessed 17 November 2021].
* Talreja, D., Nagaraj, J., Varsha, N. and Mahesh, K., 2017. Terrorism Analytics: Learning to Predict the Perpetrator. Advances in Computing Communications and Informatics (ICACCI), pp.1723-1726.
* Toure, I. and Gangopadhyay, A., 2016. Real Time Big Data Analytics for Predictions Terrorist Incidents. [Online] Academia.edu. Available at: <https://www.academia.edu/59352143/A\_method\_for\_analyzing\_terrorist\_attacks> [Accessed 11 November 2021].
* Tutun, S., 2018. AI framework by using patterns and relations to understand terrorist behaviors. [online] Medium. Available at: <https://towardsdatascience.com/ai-framework-by-using-patterns-and-relations-to-understand-terrorist-behaviors-5c8378b2cd88>
* Uddin, M., Zada, N., Aziz, F., Saeed, Y., Zeb, A., Ali Shah, S., Al-Khasawneh, M. and Mahmoud, M., 2020. Prediction of Future Terrorist Activities Using Deep Neural Networks. Complexity, 2020, pp.1-16.
* Wang, L., 2005. *Support vector machines theory and applications. Springer, Berlin*. Berlin: Springer, pp.1-43.
* Zhang, Z., 2016. Missing data imputation: focusing on single imputation. *Annals of Translational Medicine*, 4.1.