Predicting the Perpetrators of Terrorist Attacks: A Machine Learning Approach

Riccardo Iacueo

December, 2022

1 Introduction

Terrorism is a global threat that affects countries and communities around the world. It is a tactic used by terrorist groups to achieve their political, religious, or ideological goals through the use of violence and fear. The impact of terrorist attacks on victims, their families, and society as a whole is devastating and long-lasting. Number of attacks has increased year by year and also the more active groups have changed over time (Figure 1). In order to combat terrorism and protect people from its effects, it is important to understand the different types of attacks and the groups behind them. The goal of this project is to classify terrorist attacks according to the groups responsible for them. By analyzing the characteristics of attacks and the groups that carry them out, we can gain insights into the motivations, tactics, and methods used by different terrorist organizations. This knowledge can help counter-terrorism efforts and prevent future attacks.

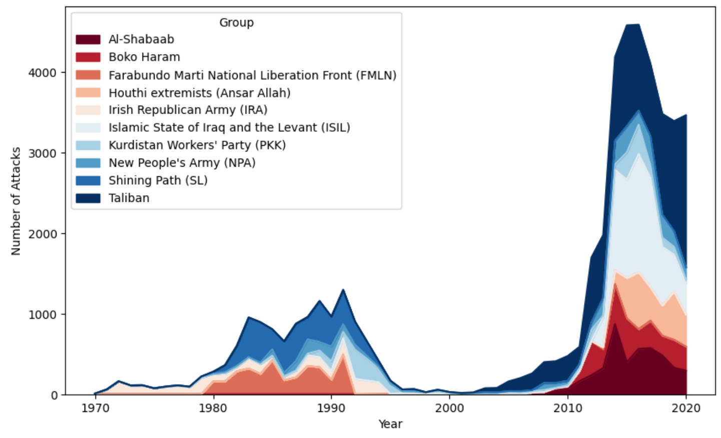

Figure 1: Top 10 terrorist Groups per Year

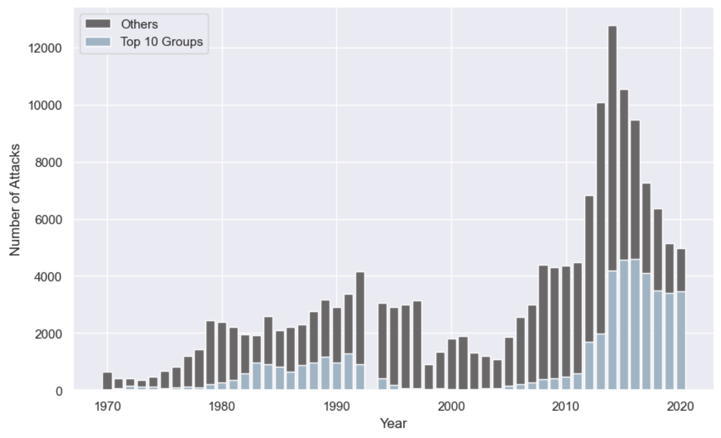

Figure 2: Top 10 Terrorist Groups' Attacks vs Total Attacks

2 Data

The dataframe considered is the Global Terrorism Database (GTD) (1). It is an open-source database managed by the **University of Maryland**. Since this is a huge dataset, we have decided to perform a first screening on the variables to be included in our final dataset; this analysis were conducted following the Codebook of the dataset (2). After having manually selected the relevant features, NaNs were evaluated for each column and covariates with a percentage greater or equal to 30% were discarded. Moreover,regarding the target variable, all the observations which values fall in the "Unknown" or "Other" category were eliminated.
To reach the final size, in order to discard irrelevant and avoid problems in model's implementation, we have decided to discard the terrorist groups which didn't carried out at the least a total number of 10 attacks. This resulted in a final dimension of approximately 79,592 observations times 25 features while the original one had

more than two hundred thousands observations and around one hundred and thirty-five features. Data for 1993 are missing because they went lost in an office move and these data have never been fully recovered, so they have been discarded from GTD. More informations about each variable included in the dataset may be found in the codebook (2).

2.1 Target variable

In this study, our target variable is the terrorist group that carried out a particular attack. The data for this variable exhibits class imbalance, with a majority of groups having carried out a relatively small number of attacks and a small number of groups having carried out a much larger number (Figure 3). Moreover, as it can be noticed by Table 1, the vast majority of groups carried out 5 attacks only. This class imbalance can have a negative impact on the performance of machine learning models trained on the data, as they may be more likely to predict the more prevalent class. To address this issue and improve the performance of our analysis, we decided to exclude groups that have carried out fewer than 10 attacks from the dataset. While this strategy helps to balance the data and reduce the impact of class imbalance on the model, it also reduces the amount of available information and excludes a significant number of groups from the analysis.

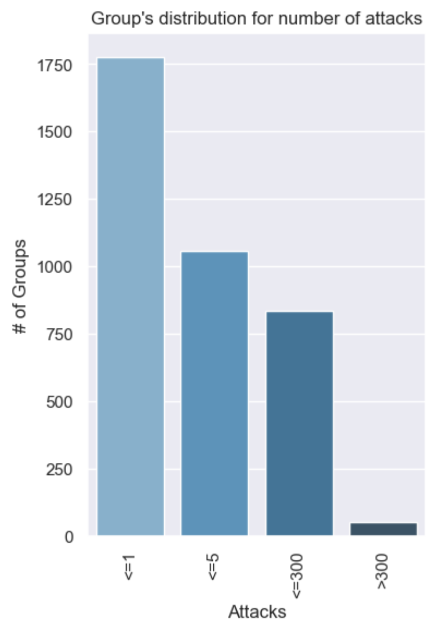

Figure 3

Table 1: Quartiles

Quartile	Attacks by Group
Q1	1
Q2	2
Q3	5

2.2 Correlation Analysis

To improve the implementation of our model, it is important to analyze the correlations between our features. To do this, we computed the correlation matrix for our dataset. We used **Pearson's R** for computing numeric-numeric associations, **Cramer's V** statistic for computing categorical-categorical associations, and the Correlation Ratio for computing categorical-numerical associations.
As shown in Figure 4, our analysis of the correlations between features did not reveal any problematic correlations. The only highly correlated features were "nkill" and "nwound", which represent the number of deaths and injuries in an attack, respectively. We expected these variables to be highly correlated, and therefore decided to keep

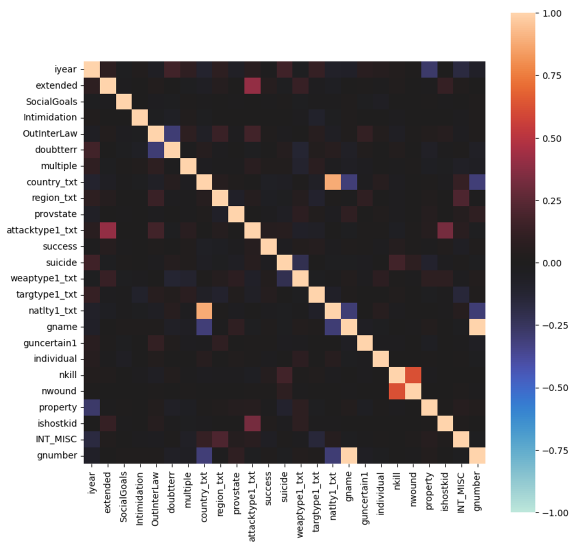

Figure 4: Correlation matrix.

them in our models to avoid discarding important information. Overall, our analysis of the correlations between features indicated that our dataset is suitable for use in our models.

3 Methodologies

In our context the classification task consist in a multiclass problem over an unbalanced dataset. When dealing with unbalanced data, where some classes are underrepresented compared to others, specialized classification techniques are needed to achieve accurate predictions. In this article, we will explore the use of Decision Trees, Bagging, Random Forests, and XGBoost, which are all powerful classification algorithms that can handle unbalanced data, for predicting the group responsible for global terrorism attacks. In this section, we will deeply discuss the models implemented, tuning techniques and metrics.

3.1 Models

Given the structure of our dataset, the models implemented falls in the category of Classification And Regression Trees: (CART) and Enseble Methods. In particular constructing four different models Classification Trees, Bagging, Random Forests and XGBoost Classifier.

3.1.1 Decision Trees, Bagging, Random Forest

The first model that we implemented is a decision tree. The reason for being the simplest and most direct choice for our dataset is that it's a simple tool for classification and can handle large datasets with many features as in the GTD. This will provide us with a good starting point for further model development and comparison. Splitting data as in Decision Trees may lead to high variability in the outcome, given a small change in the considered dataset. In order to overcome this issue, a first and straightforward approach can be to average results over different trees. This approach is called Bagged Tree (Bagging). Furthermore with Random forests, an additional random variation is added into the bagging procedure in order to create greater diversity amongst the resulting models. With Random forests, an additional random variation is added into the bagging procedure in order to create greater diversity amongst the resulting models.

3.1.2 XGBoost Classifier

The XGBoostClassifier is a class in the XGBoost library that implements the gradient boosting algorithm for classification problems. It is a type of ensemble learning method, which involves training multiple models and then combining their predictions to produce a final result. The XGBoostClassifier uses decision trees as its base learners, and trains multiple models by optimizing an objective function that measures the accuracy of the predictions. The final prediction is then made by combining the predictions of all the decision trees in the ensemble.

3.2 Tuning

To fine-tune the decision tree, we first fitted the tree to the data without specifying any parameters. Then, we used Gini accuracy and Entropy to tune the tree's maximum depth, selecting the depth that maximized these two metrics. This allowed us to find a maximum depth of 22, which provided the highest possible accuracy without overfitting the data. To optimize the Bagging and Random Forest models, we used grid search with stratified cross validation and the F1 macro score as the evaluation metric. We chose the F1 macro score because it is a balanced measure of performance that takes into account the imbalanced nature of the data. The F1 macro score calculates the F1 score for each class independently, and then takes the average across all classes, ensuring that the score is not dominated by the performance on the majority class and that the performance on each class is equally weighted. After running the grid search, we found that the optimal number of trees for the bagged tree is 11, while for Random Forest we achieved 30 as maximum depth for each tree. These values produced the highest F1 macro scores on the validation set, indicating that they were the best performing number of trees for our dataset. Due to limitations in computational power, we were unable to use grid search to tune the XGBoost model in a reasonable amount of time. Therefore, we manually tuned the model and decided to use 300 decision trees as the hyper-parameter for the XGBoost.

3.3 Metrics

To evaluate our multiclass classification models, we used precision, recall, and F1 scores to evaluate each category individually. To assess the overall performance across all categories, we used both accuracy and macroaverage. Macroaverage first calculates the precision, recall, and F1 scores for each category separately, and then takes the average of these scores across all categories. This allows us to assess the model's performance on each individual class, as well as its overall performance on the entire dataset.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 * (Precision * Recall)}{Precision + Recall}$$

In order to assess the effectiveness of our models, we employed stratified k-fold cross-validation to account for the imbalanced nature of the target variable in our dataset. This technique preserves the proportion of each class in each fold, allowing us to effectively handle imbalanced data where some classes have significantly more instances than others. We repeated the process of training and evaluating each algorithm using stratified k-fold cross-validation five times, and calculated the mean of the output to determine the cross-validation accuracy. By doing so, we were able to obtain more reliable performance estimates for our unbalanced dataset and confirm that the model was not biased towards the majority classes.

4 Results and Variable Importance

To evaluate the performance of each model, we employed the metrics outlined in Section 3.3 using a macro average to mitigate the potential influence of class imbalance on the results. However, we also computed the metrics using a weighted average, which takes into account the frequency of each class in the dataset by assigning higher weights to more prevalent classes. This can provide additional insight into the model's performance on the most frequently occurring categories. The weighted average is computed by using the number of available observations for each class as weights, meaning that classes that appear more frequently in the data contribute more to the computation of the metric.

Table 2: Results from Cross Validation

	Precision		**Recall**		**F1-Score**	
Model	Macro Average	Weighted Average	Macro Average	Weighted Average	Macro Average	Weighted Average
Decision Tree	0.35	0.64	0.38	0.55	0.35	0.58
Bagging	0.39	0.65	**0.44**	0.56	0.40	0.59
Random Forest	**0.45**	**0.67**	0.42	**0.66**	**0.42**	**0.66**
XGBoost Classifier	0.41	**0.67**	**0.44**	0.60	0.41	0.63

As we can see from Table 2 the best performing model is **Random Forest**, with a small gap between Bagging and XGBoost Classifier. Even if the results are not so satisfying we can attribute the poor performance to the high unbalance in data. In fact, by looking at the Weighted Average for each metric, each of them is higher than the correspondent macro average. This means that the most frequent class were well predicted, while the poor performance is given by the less frequent classes.

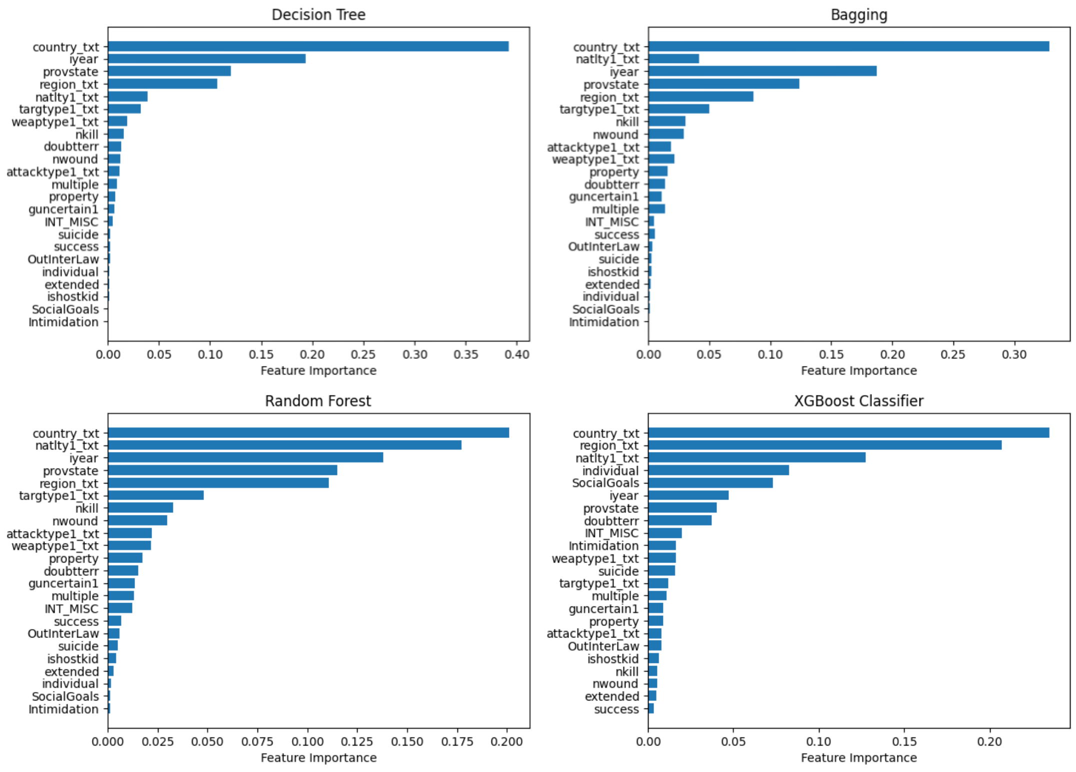

Figure 5: Variable importance for each model.

In our analysis of the variable importance for our four models, we found that most variables had similar importance across the models. However, the XGBoost Classifier showed the greatest differences in variable importance, with some variables being considered more important than in the other models. Two variables in particular, "SocialGoals" and "individual", stood out as being important in the XGBoost model.

5 Conclusions

In this study, we sought to identify similarities in the operational strategies of various terrorist organizations in order to develop models that could classify terrorist attacks based on the group that perpetrated them. To do so, we employed a combination of machine learning algorithms and analysis of attack characteristics, including type of attack, target, and number of fatalities and injuries. Among the models we tested, Random Forest proved to be the most accurate. However, we observed a decline in performance as the number of attacks attributed to a given group decreased. Both XGBoost and bagging demonstrated similar performance to random forest. Results from a weighted average metric indicated that all of the models were more effective at classifying groups with a higher frequency of attacks in the dataset. These results come from stratified 5-fold Cross Validation procedure through which we were able to evaluate the stability and performance of each model.
It is likely that improved performance could be achieved with increased computational resources, particularly for the XGBoost Classifier, which has a reputation for strong performance when properly tuned.
As an additional consideration, the use of oversampling techniques may be explored to address issues of imbalanced data and potentially enhance model accuracy for practical application.

References

[1] START (National Consortium for the Study of Terrorism and Responses to Terrorism). (2022). Global Terrorism Database 1970 - 2020 [data file]. https://www.start.umd.edu/gtd.

[2] START (National Consortium for the Study of Terrorism and Responses to Terrorism). (2021, August). Global Terrorism Database codebook: Methodology, inclusion criteria, and variables. University of Maryland. https://www.start.umd.edu/gtd/downloads/Codebook.pdf.

[3] G. LaFree and L. Dugan, "Introducing the global terrorism database," Terrorism and Political Violence, vol. 19, no. 2, pp. 181–204, 2007.

[4] James, G., Witten, D., Hastie, T., Tibshirani, R. (2013). An introduction to statistical learning: With applications in R (1st ed.). Springer.