**Prediction of Unidentified Terrorist Attacks   
in Pakistan**

**Riddhimoy Ghosh**

University of Nebraska at Omaha

**Context**

*Machine Learning tools are highly beneficial for investigators as they provide valuable insights into the underlying mechanisms of terrorism. Here we work with a dataset on terrorist attacks within Pakistan by multiple terrorist groups. Some terrorist attacks have been traced back to the source organizations by their claims/ other witness identifications while the others have remained untraced. This project will attempt to fit a prediction model to predict the terrorist organizations behind unclaimed terrorist attacks based on historical data around claimed and/or identified terrorist attacks.*

**Research Questions**

* The best prediction possible of which terrorist organization committed each of the unidentified attacks in the data set.

**Highlights and Comments on the Data Set Used:**

* The dataset under consideration contains 12,237 terrorist records (identified by a unique event identifier *eventid*) of terrorist attacks in Pakistan, initiated respectively by various terrorist groups, in between 2007-2018.
* In this project we focused trying to predict the group name of the terrorist organization (*gname*) that may have initiated a unique terrorist attack whose initiator is *unknown*. We observe that approximately 75% unique attack event records’ initiator is unknown. So, our task is to use the 25% identified attack event records to develop a Machine

A picture containing text, screenshot, plot

Description automatically generated Learning Model that can be used to predict the 75% of the unknown attack initiators.

* In this regard we also note that a particular attack event can be ‘claimed’ (1), ‘unclaimed’ (0) or ‘unknown’ (-9) (from *claimed* column). For certain attack events the initiator group had claimed to be perpetrator and that’s how the group behind it is identified; in some cases when an attack is unclaimed, we still get to know the perpetrator group based on witness reports or enquiries.

A blue rectangle with white text

Description automatically generated with medium confidence

As expected, none of the unknown group events had any claimants and for individual terrorist groups the percentage of claimed attacks vary. For e.g., based on available records 'tehrik-i-taliban pakistan (ttp)' claimed ~57% of the attacks they initiated while 'baloch republican army (bra)' had claimed ~95% of the attacks they had initiated.

* From the information provided along with the dataset we get to know that ‘-9’ represents missing values. We also note that there are non-coded missing values present in the data as well. For e.g., consider below some example columns.



We will revisit the strategy for handling the missing values in the model building section.

**Pre-Modeling Observations and Considerations:**

* From the above discussion we find that the problem at hand is multi-class classification problem where we have to predict the class label (terrorist groups) of the attack event ids with *unknown* groups (~75%). We have only ~25% of all the available events to build our model upon and under the assumption that each unknown attacks have been initiated by one of the known groups from the data.
* We also note that , there are 78 unique terrorist groups (candidate class labels for our multiclass prediction model) who have initiated the known attack group events; out of which ~45% are by 'tehrik-i-taliban pakistan (ttp)' , ~10% are by 'baloch republican army (bra)', ~5% are by 'baloch liberation front (blf)' and the rest of the 75 terrorist groups have individual attack contributions of less than 5% with majority having less than 1%.
* Based on the above 2 observations, we have a huge class imbalance problem here – we do not have enough example true events to have our ML model learn and discern between the event initiations of individual terrorist organizations. Due to relative abundance of event examples from 'tehrik-i-taliban pakistan (ttp)', the model would tend to overfit on this terrorist group and would be susceptible to attribute ‘non-ttp’ *unknown* attacks as ‘ttp’ events.
* Keeping the above class imbalance problem in mind in a multi-class classification setting, we could consider the below options:
  + **Reduce the number of classes:** Trying to group some of the less represented classes together into broader categories, to reduce the number of classes that need to be predicted. This can help to improve the overall performance of the model and can also simplify the problem.  
     One of the options here to turn this multi-class into binary classification problem by keeping 'tehrik-i-taliban pakistan (ttp)' as an individual class label (‘1’) and pooling all the other groups together to form the (‘0’) class.
  + **Ensemble Learning:** Training multiple models on different subsets of the data, and then combine their predictions. This can help to improve the overall performance of the model and can also help to address the issue of imbalanced classes.
  + **Undersampling/ Oversampling/ Class Weighting Techniques:** Randomly select a subset of the majority class to balance the dataset; Duplicating minority class samples to increase their representation in the dataset and assigning higher weight to minority classes during training respectively. These are sensitive maneuvers and require advanced knowledge.

**Final Modeling Approach:**

* Based on the above considerations, we went forward to approach this problem as a multi-classification with reduced number of classes (4). We kept the top-3 classes by frequency: 'tehrik-i-taliban pakistan (ttp)', 'baloch republican army (bra)’, ‘baloch liberation front (blf)' as 3 individual classes and pooled the others under ‘others’ category.   
  We considered reducing the above to a binary classification problem by keeping 'tehrik-i-taliban pakistan (ttp)' (1) pooling all the other groups to ‘others’ (0) category, but we did not want to oversimplify the problem statement in that regards and preserve the multiclass characteristics to best of our ability.
* We decided to go with Random Forest Classifier as our ensemble modeling candidate. We did not try to experiment with many different classification algorithms but focus more cross validation and hyperparameter tuning to finalize a model that can generalize well to unseen data.
* In this regard we experimented with 2 different approaches of multiclass modeling:
  + A single Random Forest Mutli-Class classifier model
  + Training 4 different binary Random Forest classifiers and then using a voting ensemble to make the final predictions. Here we use soft voting, where final class probabilities are obtained by averaging the predicted probabilities from each individual classifier.  
      
    Ensembling binary classifier through soft voting is more resource consuming approach and we were curios if it provided some gain over having a single multi-class classifier.

**Modeling Details and Specification:**

* **Variable Selection:**
  + We went forward with putting in those variables in the model that were defined in the data dictionary. We kept the *eventid* variable aside for mapping the model predictions with the events.
  + In cases where we had 2 categorical columns like *region* and *region\_txt* that represented the same information, we dropped the \_txt columns that had text mapping of the numeric categories.
  + We used 26 predictors from the original dataset columns and the modified response variable *target (*based out of the *gname* column*)* *\* Please see Appendix for Final Variable List*
* **Pre-processing Steps:**
  + We used the sckit-learn pipelines to streamline our preprocessing steps. In this way we could fit (transform) our pre-processer on the training set and then use the same to transform the test set.
  + The categorical pipeline imputed the missing values with the modes and then one hot-encoded categorical variables.
  + The numeric pipeline imputed the missing values with median and scaled the numeric variables with max absolute scaler (we then combined the individual pipelines to a single column transformer)
* **Hyper-parameter Tuning:**

We tuned the hyper-parameters of the Random Forest Model *‘n\_estimators’, ‘min\_samples\_split’ and ‘max\_features’* through Randomized Search technique over a defined parameter grid with 5-fold cross validation for each combination.

* **Final Model and Accuracy Score:**

The tuned hyper-parameter values for the final Random Forest Classifier:

* + - 'n\_estimators': 450
    - 'min\_samples\_split': 10
    - 'max\_features': 'sqrt'

The cross validated Accuracy Score for the model is ~67.30% and the cross validated Accuracy Score for the Ensemble using voting classifier is ~66.95%.

**Modeling Outcomes:**

* **Feature Importance:**

A picture containing text, screenshot, plot, colorfulness

Description automatically generated

* **Prediction of the Unidentified Attacks Events:**
  + **From Single Random Forest Classifier:**



* + **From Ensemble Using Voting Classifier:**



* **Percentage of matching class predictions: ~97.16%**

**Challenges, Learnings and Takeaways:**

* I felt the toughest decision I had to make is how to approach this multi-class classification problem where there was a huge class imbalance, presence of numerous sparse classes and the training set being way small relative to the test set whose class labels we strove to predict. There is probably no 'perfect' approach to tackling these kinds of problems and depends highly on the context and goals at hand. Reducing the problem to a binary classification problem with trying to predict 'tehrik-i-taliban pakistan (ttp)' vs all others could have directly addressed the class imbalance issue, but I felt that it morphed the original research question. I really hope to improve on this directional decision-making skill as I complete more projects and brainstorm about the results delivered.
* The other important aspect was finding the right Machine Learning model from our toolbox that can produce good results in the given settings, coupled with the sufficient hyperparameter tuning and cross-validations applied. Due to time constraints here, we could not explore boosting algorithms like xgboost that could potentially give better results compared to random forest.

**Appendix:**

* **Model Variable List:**

