Fellowship.AI Challenge

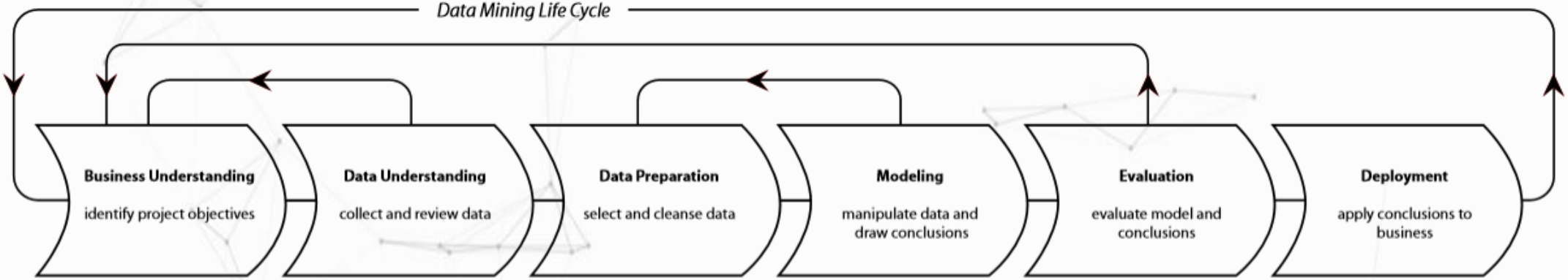
Predicting Terror Group Culpability

Contents

Executive Summary

- The goal of this project was: **to predict the culpability of terrorist groups** for attacks based of the attributes of that attack in order to help build evidence-based anti-terror policy in areas where this could help reduce attacks most
- The **analysis focused on the Middle East & North Africa** as it was the region with the highest number of unknown attacks and the highest number of causalities from terror attacks
- Geographic variables like **country were the largest discriminating factor** for groups due to the locational specificity of many groups, though other variables such as **nationality of victims and method of attack were also able to discriminate effectively** between co-located groups
- **Non-linear ensemble models worked best** for this dataset, with the top-performing models (Gradient Boosting and Bagging Classifiers) achieving a **precision of ~96%**. These were able to perform robustly even against unseen evaluation sets
- **The model was deployed on recent unknown attacks in the Middle East** and could be used to attribute culpability to groups in the area and thus feed evidence-based anti-terror policy

This analysis uses the CRISP-DM methodology to help structure the data insights process



Determine Business Objectives
Background
Business Objectives
Business Success Criteria
(Log and Report Process)

Assess Situation
Inventory of Resources, Requirements, Assumptions, and Constraints
Risks and Contingencies
Terminology
Costs and Benefits
(Log and Report Process)

Determine Data Mining Goals
Data Mining Goals
Data Mining Success Criteria
(Log and Report Process)

Produce Project Plan
Project Plan
Initial Assessment of Tools and Techniques
(Log and Report Process)

Collect Initial Data
Initial Data Collection Report
(Log and Report Process)

Describe Data
Data Description Report
(Log and Report Process)

Explore Data
Data Exploration Report
(Log and Report Process)

Verify Data Quality
Data Quality Report
(Log and Report Process)

Data Set
Data Set Description
(Log and Report Process)

Select Data
Rationale for Inclusion/ Exclusion
(Log and Report Process)

Clean Data
Data Cleaning Report
(Log and Report Process)

Construct Data
Derived Attributes
Generated Records
(Log and Report Process)

Integrate Data
Merged Data
(Log and Report Process)

Format Data
Reformatted Data
(Log and Report Process)

Select Modeling Technique
Modeling Technique
Modeling Assumptions
(Log and Report Process)

Generate Test Design
Test Design
(Log and Report Process)

Build Model Parameter Settings
Models
Model Description
(Log and Report Process)

Assess Model
Model Assessment
Revised Parameter
(Log and Report Process)

Evaluate Results
Align Assessment of Data Mining Results with Business Success Criteria
(Log and Report Process)

Approved Models
Review Process
Review of Process
(Log and Report Process)

Determine Next Steps
List of Possible Actions
Decision
(Log and Report Process)

Plan Deployment
Deployment Plan
(Log and Report Process)

Plan Monitoring and Maintenance
Monitoring and Maintenance Plan
(Log and Report Process)

Produce Final Report
Final Report
Final Presentation
(Log and Report Process)

Review Project
Experience Documentation
(Log and Report Process)

https://decisionstats.files.wordpress.com/2011/10/12345.png?w=840

Before starting the analysis, it is crucial to have a clear understanding of the how this work could add real-world value and set clear goals for the project in line with this:

What are the real-world issues and applications:

- Terrorism is one of the most salient political issues of the 21st century. While some terror groups claim responsibility for attacks, others remain anonymous, making it difficult to attribute culpability for terror incidents. This can in turn make it difficult to set policy agendas and dedicate resources optimally to combating the most dangerous groups.
- The **Global Terrorism Database (GTD)*** records information and attributes on both known and unknown terror attacks across the world. Using this database to build a model which can train (using the known incidents) in order to predict the unknown incidents would therefore be highly valuable as a tool to help attribute culpability for attacks to specific groups. It is hoped that the results of this model could therefore help build policy objectives and allocate anti-terrorism resources in a more evidence-based way. It may also have value for helping understand past historical events for academic and social-justice purposes.
- While the GTD is a *global* dataset however, different regions of the world experience and suffer from terror attacks differently. It is therefore worth taking into consideration that a 'once-size-fits-all' approach may not be best for this problem, with different situations on the ground and different governments and regional bodies having different requirements. Focusing on a single, highly affected region may therefore give the highest value (especially if that region has many unknown attacks which could be attributed).
- The goal of this project will therefore be:

To predict the culpability of terrorist groups for attacks based of the attributes of that attack in order to help build evidence-based anti-terror policy in areas where this could help reduce attacks most

How can this be translated into technical goals:

- In technical terms this will be a **multi-class classification** problem as we want to attribute responsibility for attacks to one of many distinct groups.
- As the GTD records are relatively small (in comparison to say website click-times datasets of potentially billions of records), **shallow algorithms** are likely to be a good first port of call – both in terms of being able to generalize well and in terms of reduced hardware needs. However depending on the scale and real-world budget of this project, deep learning and powerful server architectures may still be a viable option.
- Considering our real-world goals, it is important to also have a relevant performance metric. Because of the potentially deadly implications of informing policy based on this model (e.g. military operations may be directed towards a specific group) it is important that culpability is only attributed to groups when we are confident they are likely to be the true perpetrator. In other words the cost of blaming a group wrongly (False Positive) is a lot worse than not identifying a group that is in fact guilty (False Negative). For this reason, amongst others, **Precision** seems an intuitively good performance metric, though more will be discussed on this later.
- Moreover, considering the real-world goal of helping set evidence-based policy, then there should not be a massive constraint on the system performance of the model (i.e. attributing responsibility in seconds rather than minutes should not make much difference to setting good policy), though variable server/hardware costs of computationally expensive models may want to be considered.
- Finally we need to remember this is 'real-world' data and thus may have quality and definitional issues. In other words the data itself may not always be correct. Moreover the definition, constitution and behaviour of these groups is likely to change over time, suggesting that models would need to be evaluated carefully and retrained regularly to stay relevant to the real-world goal.

* https://www.start.umd.edu/gtd/

The first thing to do is understand the dataset itself

181,691 attacks (records)…

135 attributes (fields)…

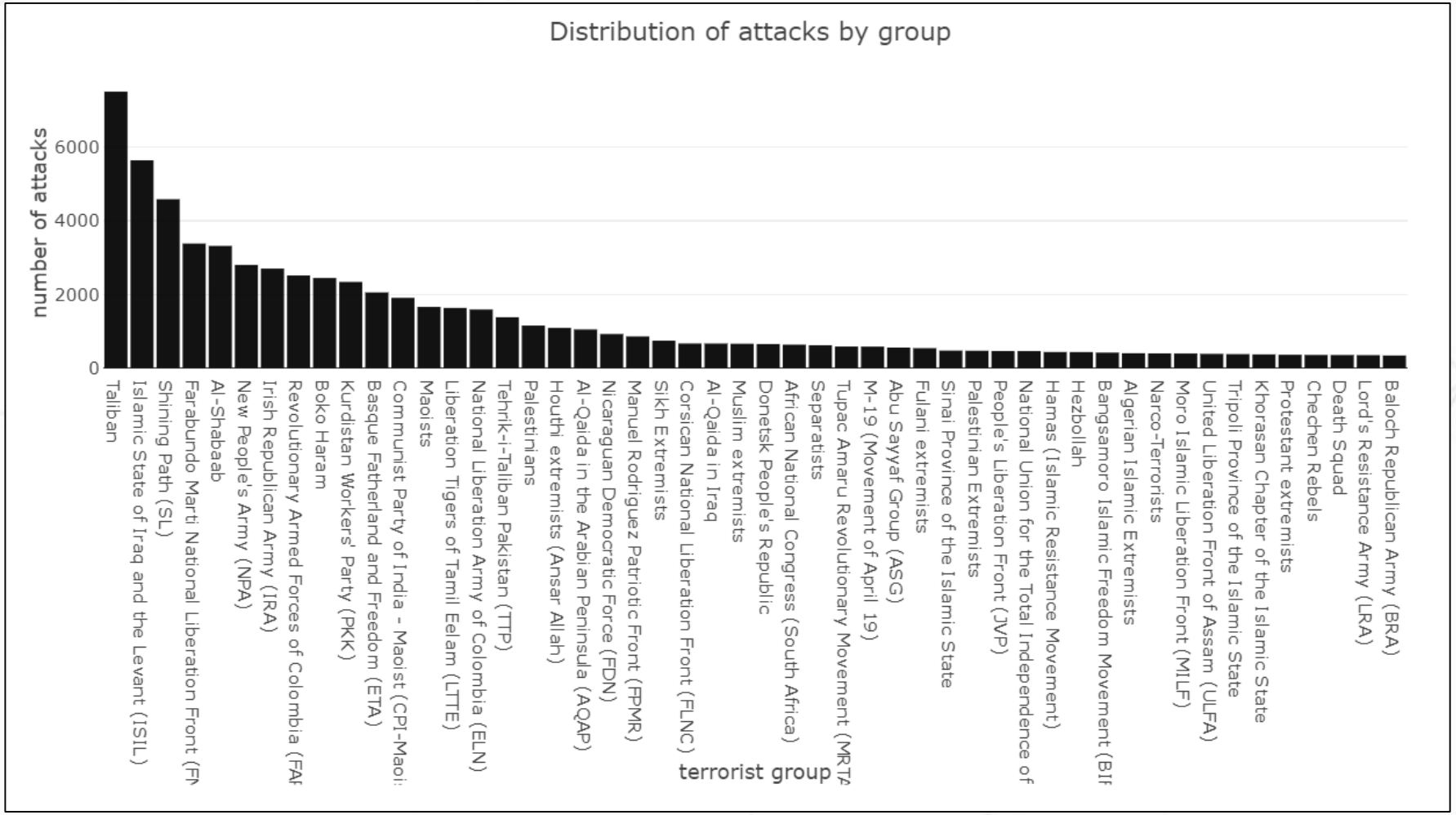
3556 unique groups (classes)…

45% unknowns (unlabelled instances)

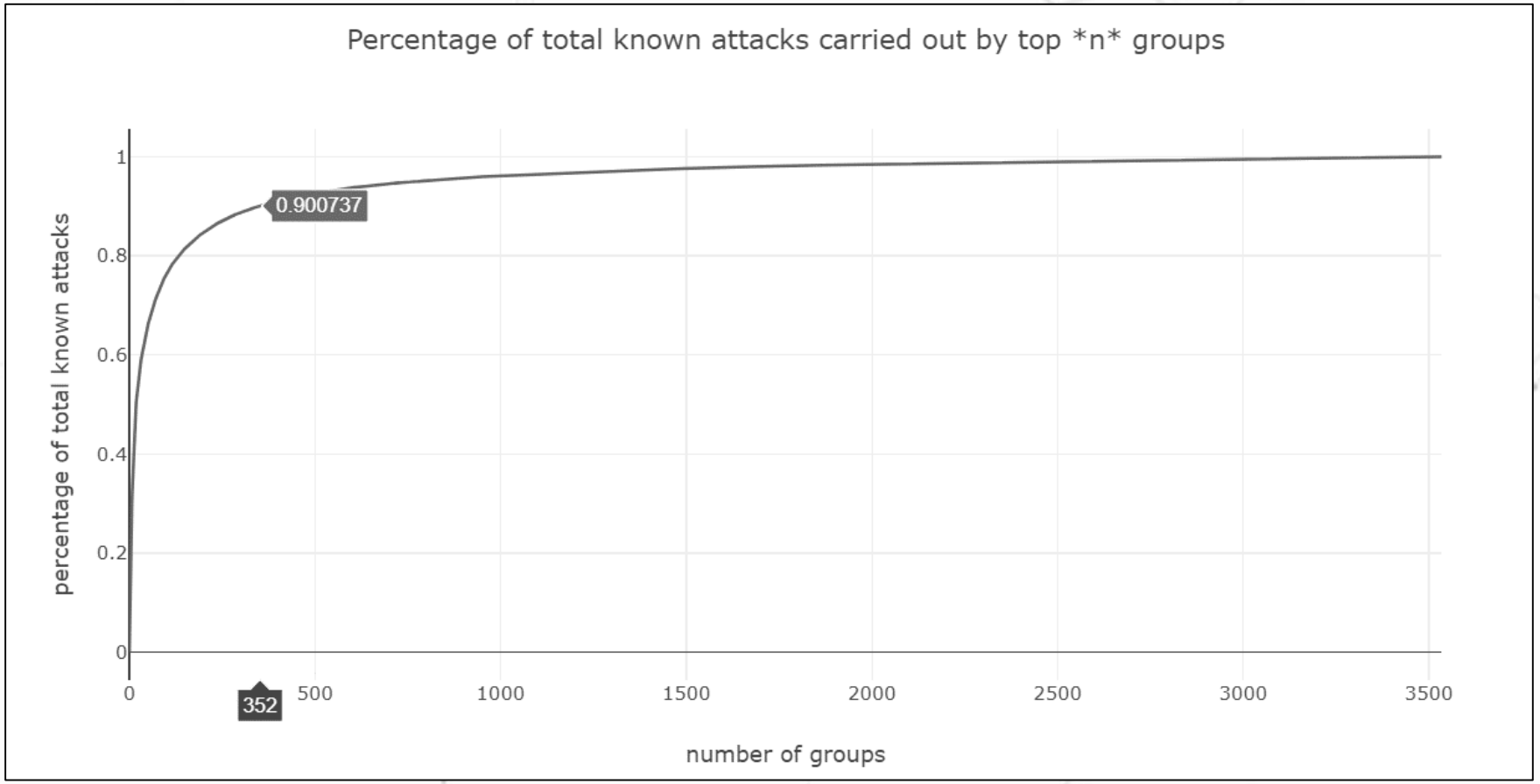
All fields in a readable format:

	field_batch_0	field_batch_1	field_batch_2	field_batch_3	field_batch_4	field_batch_5	field_batch_6	field_batch_7	field_batch_8
0	(eventid, int64, 0.0)	(specificity, float64, 0.0)	(attacktype2, float64, 0.97)	(targsubtype2_txt, object, 0.94)	(gname2, object, 0.99)	(claimmode2, float64, 1.0)	(weaptype3_txt, object, 0.99)	(propextent, float64, 0.65)	(ransompaidus, float64, 1.0)
1	(iyear, int64, 0.0)	(vicinity, int64, 0.0)	(attacktype2_txt, object, 0.97)	(corp2, object, 0.94)	(gsubname2, object, 1.0)	(claimmode2_txt, object, 1.0)	(weapsubtype3, float64, 0.99)	(propextent_txt, object, 0.65)	(ransomnote, object, 1.0)
2	(imonth, int64, 0.0)	(location, object, 0.69)	(attacktype3, float64, 1.0)	(target2, object, 0.94)	(gname3, object, 1.0)	(claim3, float64, 1.0)	(weapsubtype3_txt, object, 0.99)	(propvalue, float64, 0.79)	(hostkidoutcome, float64, 0.94)
3	(iday, int64, 0.0)	(summary, object, 0.36)	(attacktype3_txt, object, 1.0)	(natlty2, float64, 0.94)	(gsubname3, object, 1.0)	(claimmode3, float64, 1.0)	(weaptype4, float64, 1.0)	(propcomment, object, 0.68)	(hostkidoutcome_txt, object, 0.94)
4	(approxdate, object, 0.95)	(crit1, int64, 0.0)	(targtype1, int64, 0.0)	(natlty2_txt, object, 0.94)	(motive, object, 0.72)	(claimmode3_txt, object, 1.0)	(weaptype4_txt, object, 1.0)	(ishostkid, float64, 0.0)	(nreleased, float64, 0.94)
5	(extended, int64, 0.0)	(crit2, int64, 0.0)	(targtype1_txt, object, 0.0)	(targtype3, float64, 0.99)	(guncertain1, float64, 0.0)	(compclaim, float64, 0.97)	(weapsubtype4, float64, 1.0)	(nhostkid, float64, 0.93)	(addnotes, object, 0.84)
6	(resolution, datetime64[ns], 0.99)	(crit3, int64, 0.0)	(targsubtype1, float64, 0.06)	(targtype3_txt, object, 0.99)	(guncertain2, float64, 0.99)	(weaptype1, int64, 0.0)	(weapsubtype4_txt, object, 1.0)	(nhostkidus, float64, 0.93)	(scite1, object, 0.36)
7	(country, int64, 0.0)	(doubtterr, float64, 0.0)	(targsubtype1_txt, object, 0.06)	(targsubtype3, float64, 0.99)	(guncertain3, float64, 1.0)	(weaptype1_txt, object, 0.0)	(weapdetail, object, 0.37)	(nhours, float64, 0.98)	(scite2, object, 0.58)
8	(country_txt, object, 0.0)	(alternative, float64, 0.84)	(corp1, object, 0.23)	(targsubtype3_txt, object, 0.99)	(individual, int64, 0.0)	(weapsubtype1, float64, 0.11)	(nkill, float64, 0.06)	(ndays, float64, 0.96)	(scite3, object, 0.76)
9	(region, int64, 0.0)	(alternative_txt, object, 0.84)	(target1, object, 0.0)	(corp3, object, 0.99)	(nperps, float64, 0.39)	(weapsubtype1_txt, object, 0.11)	(nkillus, float64, 0.35)	(divert, object, 1.0)	(dbsource, object, 0.0)
10	(region_txt, object, 0.0)	(multiple, float64, 0.0)	(natlty1, float64, 0.01)	(target3, object, 0.99)	(nperpcap, float64, 0.38)	(weaptype2, float64, 0.93)	(nkillter, float64, 0.37)	(kidhijcountry, object, 0.98)	(INT_LOG, int64, 0.0)
11	(provstate, object, 0.0)	(success, int64, 0.0)	(natlty1_txt, object, 0.01)	(natlty3, float64, 0.99)	(claimed, float64, 0.36)	(weaptype2_txt, object, 0.93)	(nwound, float64, 0.09)	(ransom, float64, 0.57)	(INT_IDEO, int64, 0.0)
12	(city, object, 0.0)	(suicide, int64, 0.0)	(targtype2, float64, 0.94)	(natlty3_txt, object, 0.99)	(claimmode, float64, 0.89)	(weapsubtype2, float64, 0.94)	(nwoundus, float64, 0.36)	(ransomamt, float64, 0.99)	(INT_MISC, int64, 0.0)
13	(latitude, float64, 0.03)	(attacktype1, int64, 0.0)	(targtype2_txt, object, 0.94)	(gname, object, 0.0)	(claimmode_txt, object, 0.89)	(weapsubtype2_txt, object, 0.94)	(nwoundte, float64, 0.38)	(ransomamtus, float64, 1.0)	(INT_ANY, int64, 0.0)
14	(longitude, float64, 0.03)	(attacktype1_txt, object, 0.0)	(targsubtype2, float64, 0.94)	(gsubname, object, 0.97)	(claim2, float64, 0.99)	(weaptype3, float64, 0.99)	(property, int64, 0.0)	(ransompaid, float64, 1.0)	(related, object, 0.86)

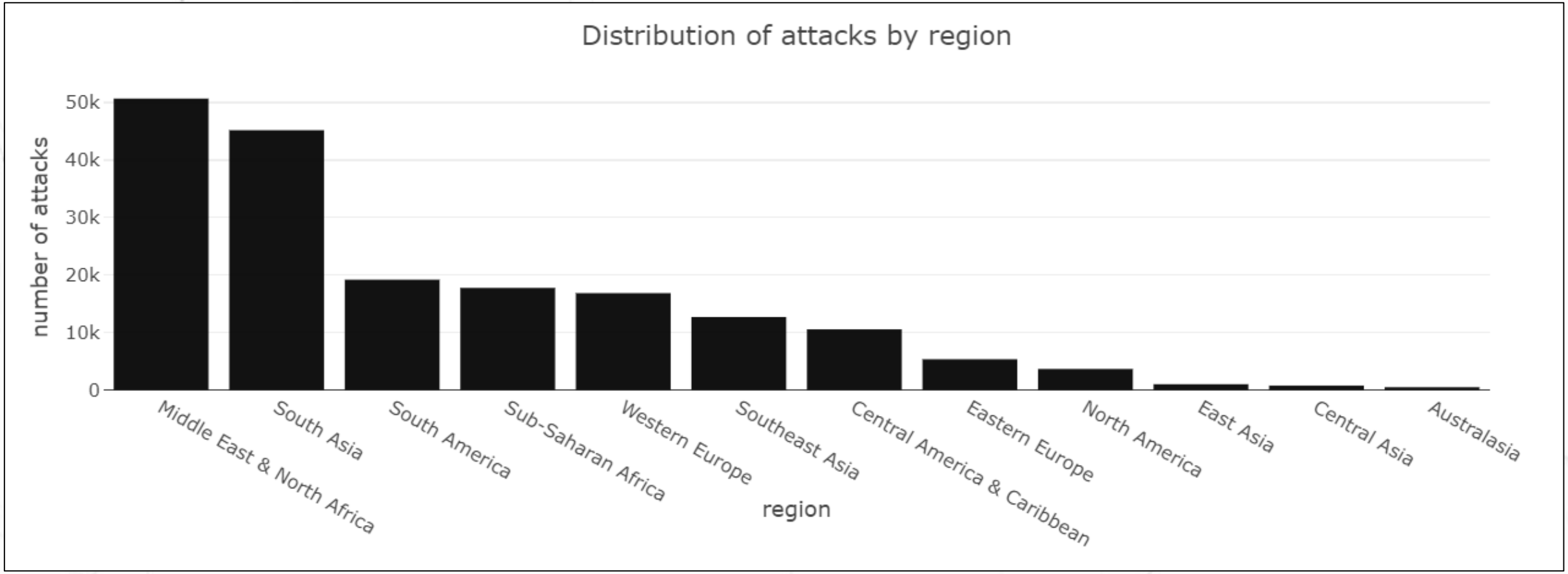
While there were more than 3500 unique terror groups represented, a small number of these groups are committing a large proportion of the attacks…



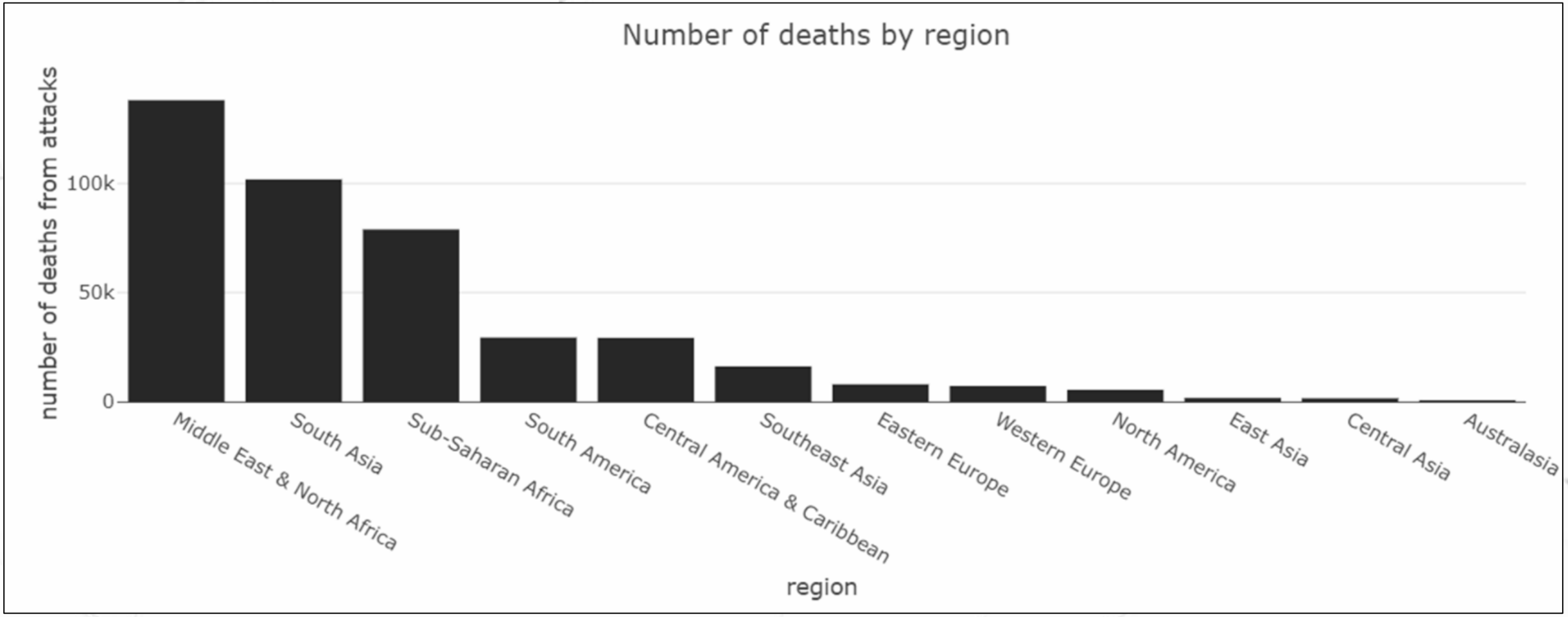
In fact more than 90% of attacks were committed by just the top 10% (~350) groups globally, suggesting that focusing analysis on these top groups could provide the most value …



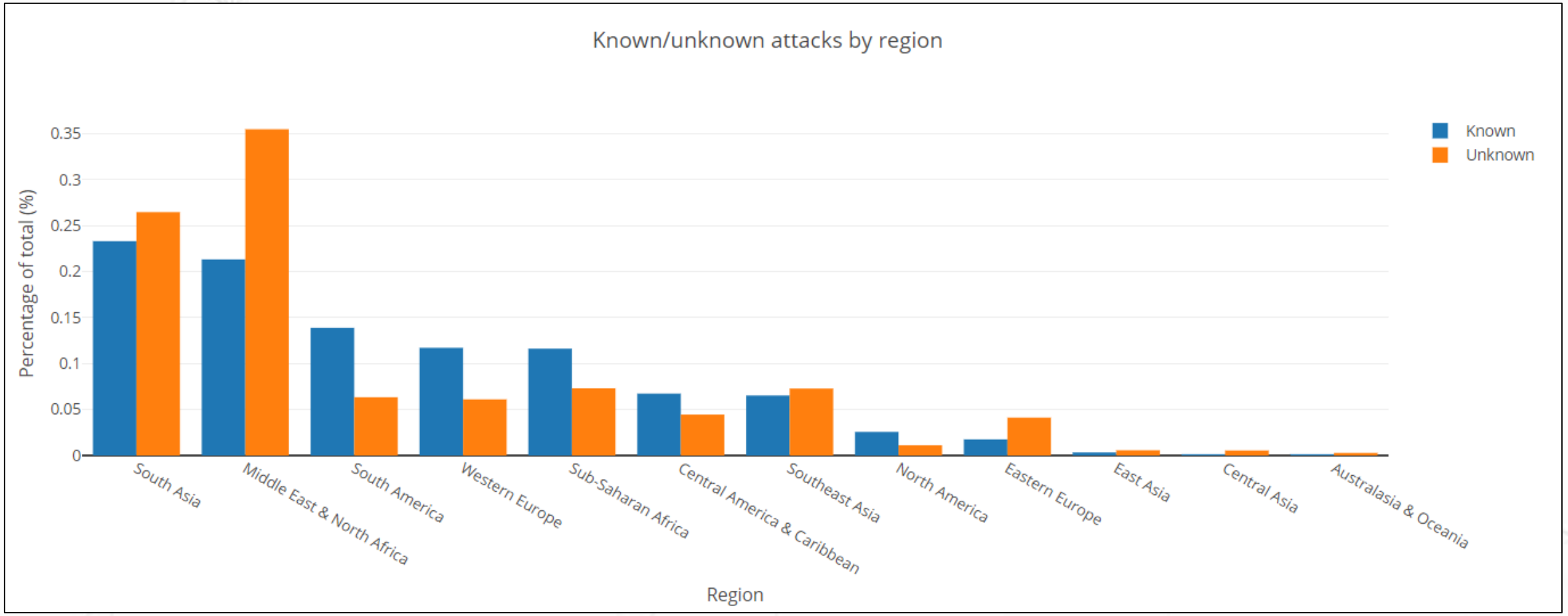
Breaking down the data regionally, it is clear that the Middle East & North Africa (followed by South Asia) has seen both the majority of attacks…



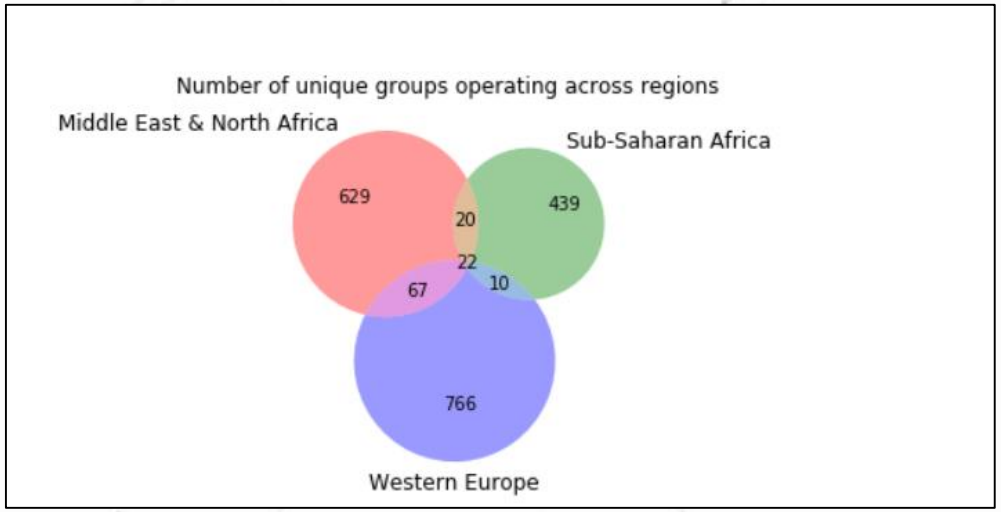
… as well as the highest causality rate from terror attacks…

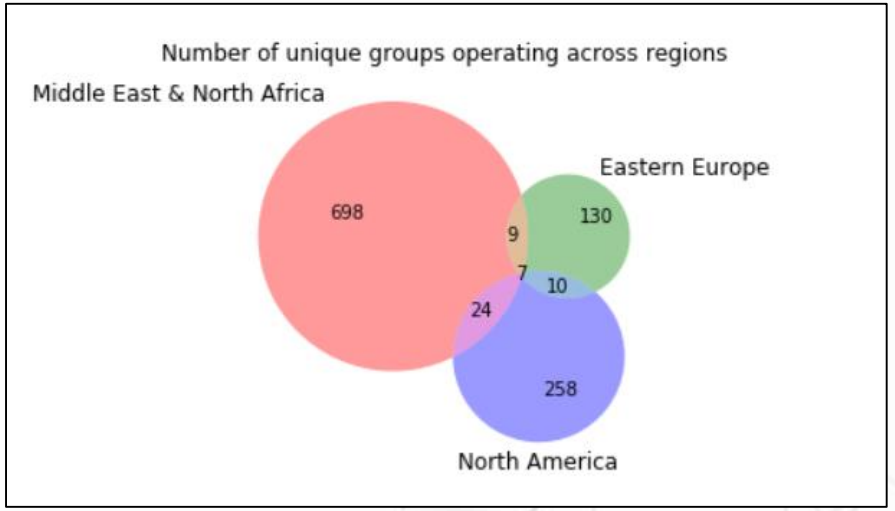


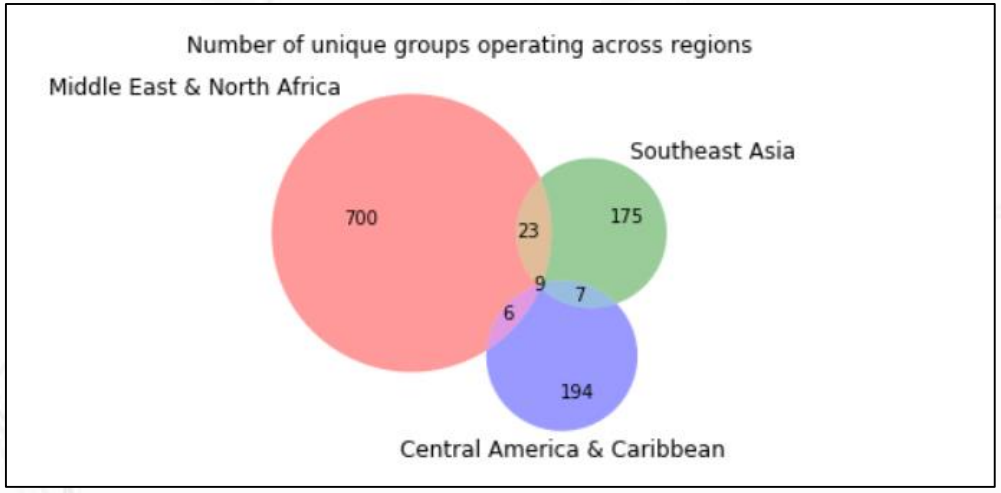
… And by far the most ‘Unknown’ attacks to date

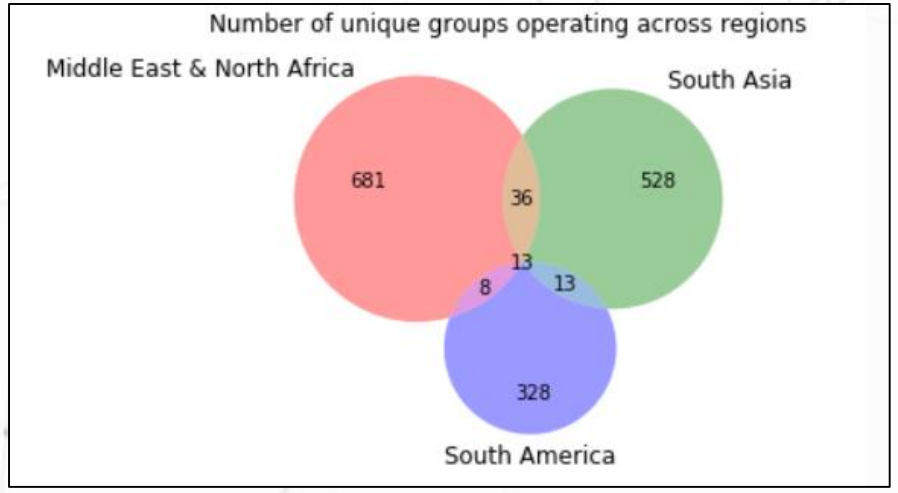


While there are some groups that operate across regions, the vast majority operate in a single region, suggesting regionally specific models to be a good approach…



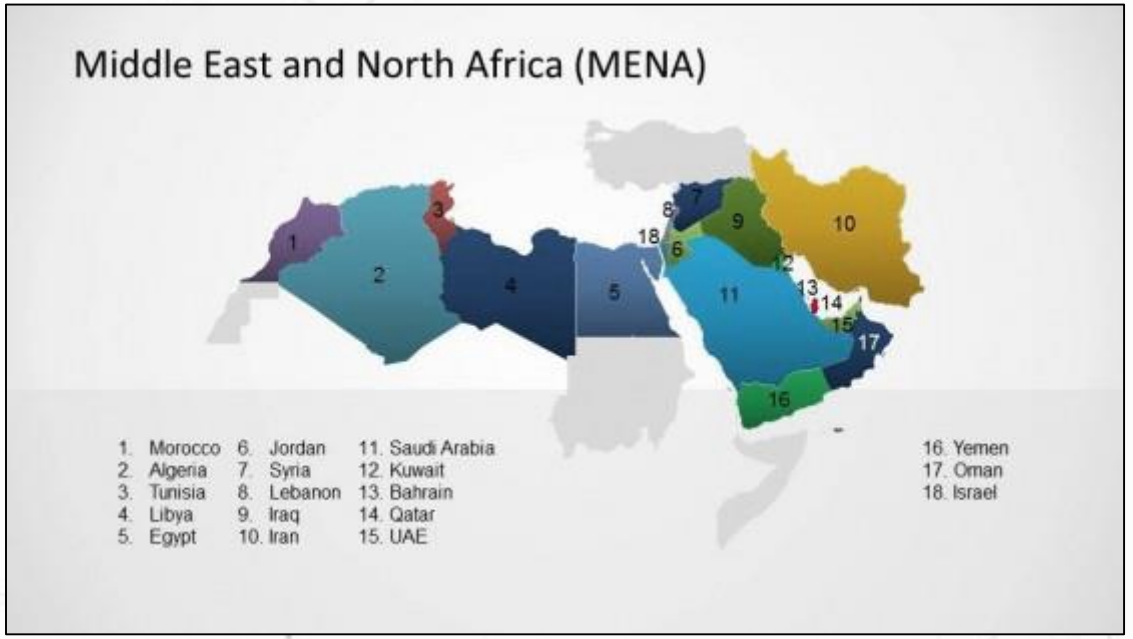






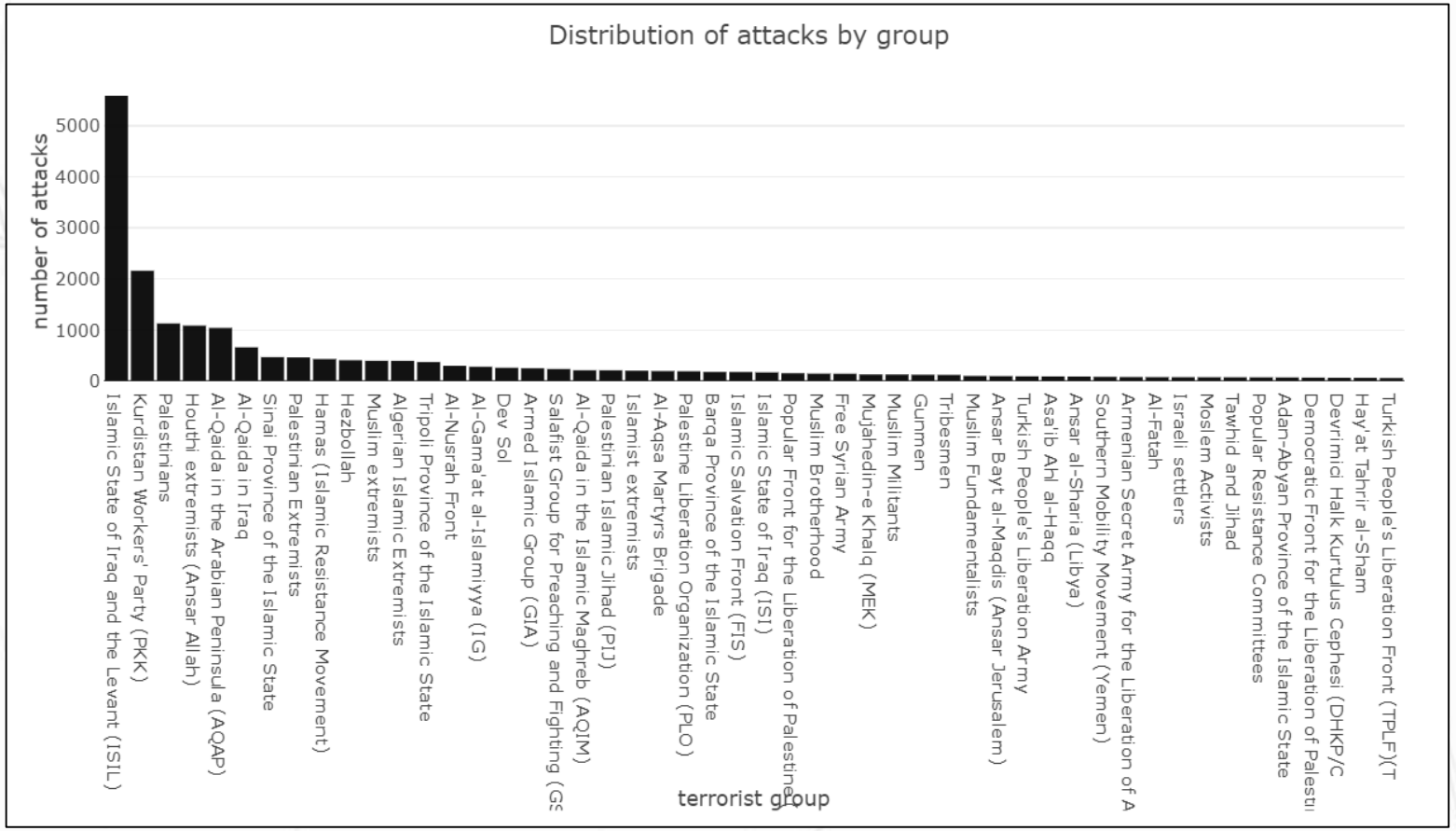
This analysis decided to focused on the Middle East & North Africa region as the highest impact zone

- Given the evidence above it seems clear the 'Middle East and North Africa' could be considered the most important region to focus analysis, both **in terms of raw number of attacks, number of casualties, and most unknown attacks (highest deployment value).** For this reason the rest of the analysis will focus only on this region.
- However, while the analysis itself will be regionally specific, the **code will be written in such a way as to allow new regions to be plugged in** (e.g. with regional specific variables acting as input parameters to model functions). In this way it is hoped that much of the code could be repurposed to build other regional models easily, and could potentially be drawn together into a global model eventually.

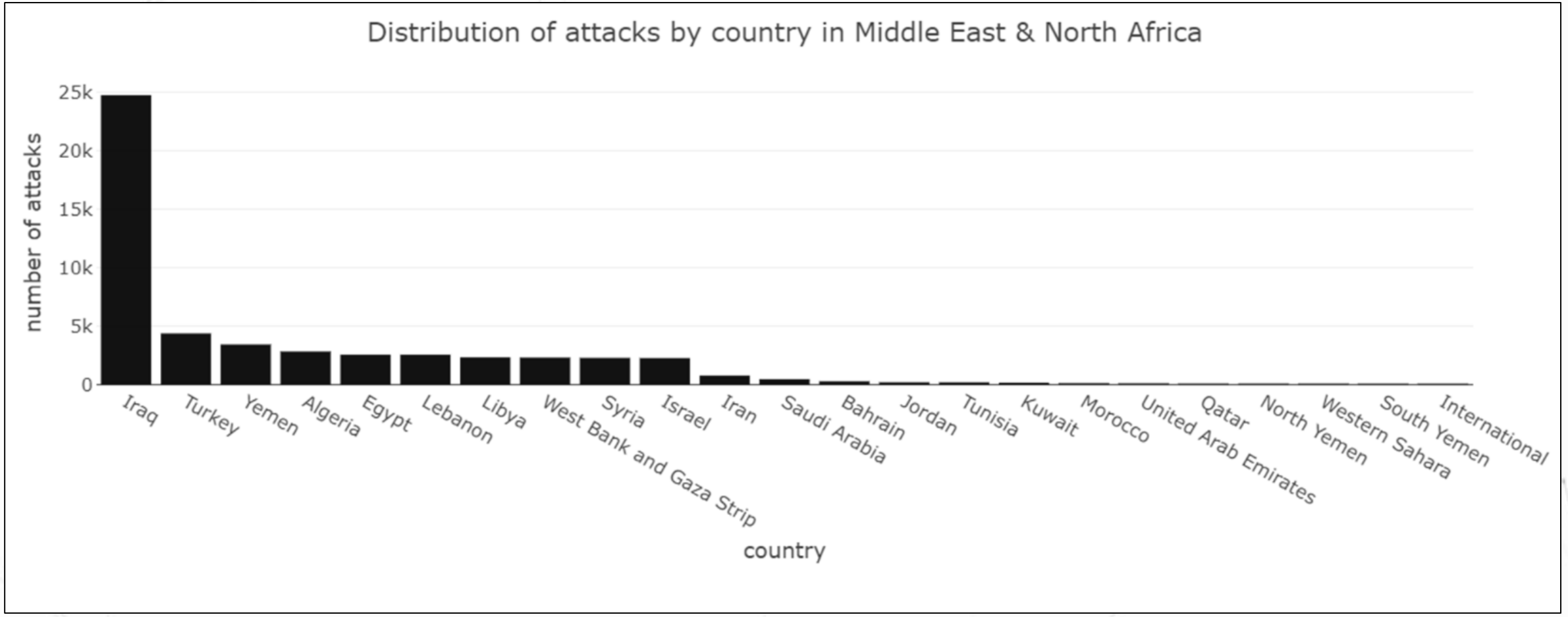


This analysis will also focus on just the most active groups in the Middle East due to their disproportioned impact

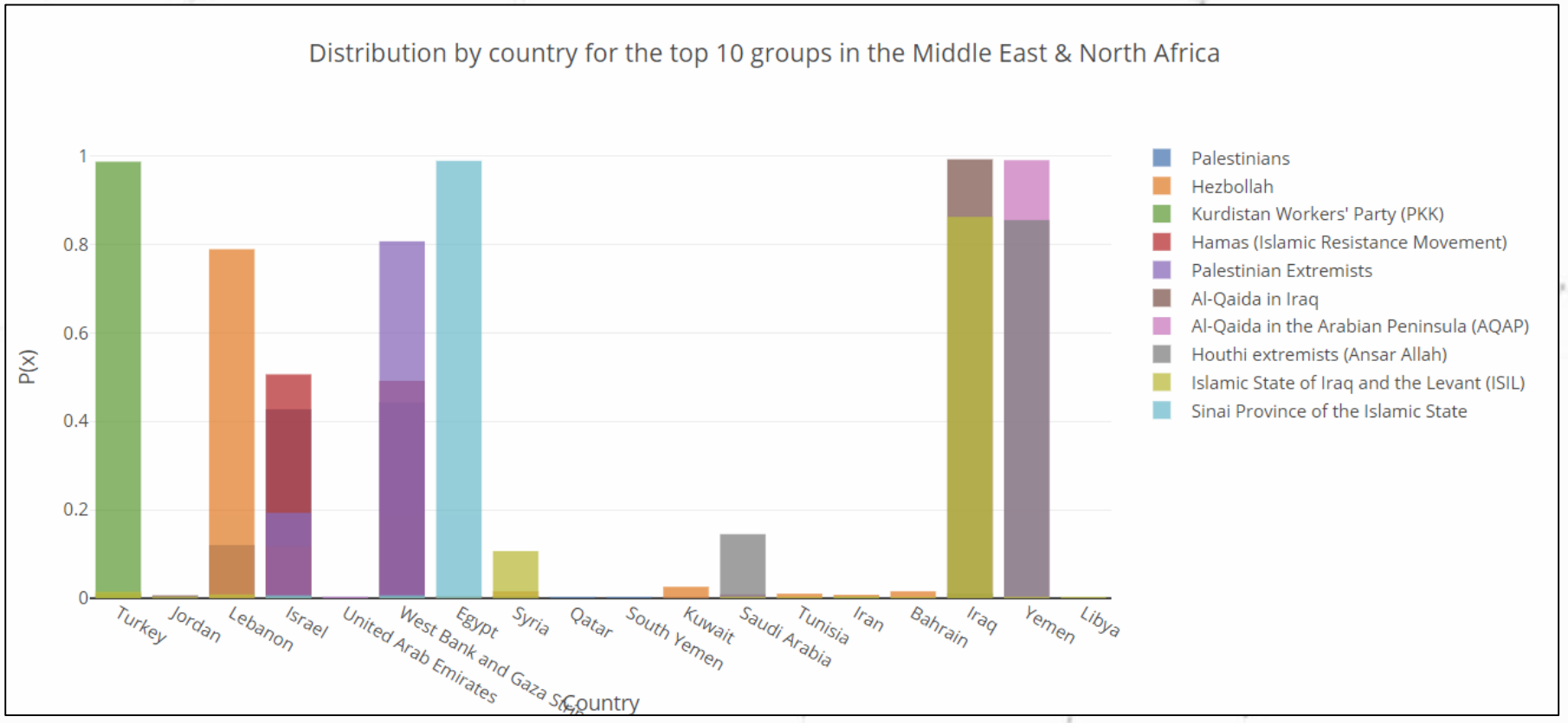
- The analysis will **also subset the number of groups under consideration to the top 20 by number of attacks**, as these have disproportionality the biggest impact and thus value in identifying.
- However it is fair to say this is also to reduce the problem complexity and in real deployment we may well want to extend this out to the long-tail of smaller groups. This is discussed further in the conclusion.
- It should also be noted that 'generic' groups (e.g. 'Gunmen') will not be considered here to help focus on identifying specific groups which could be targeted in policy making



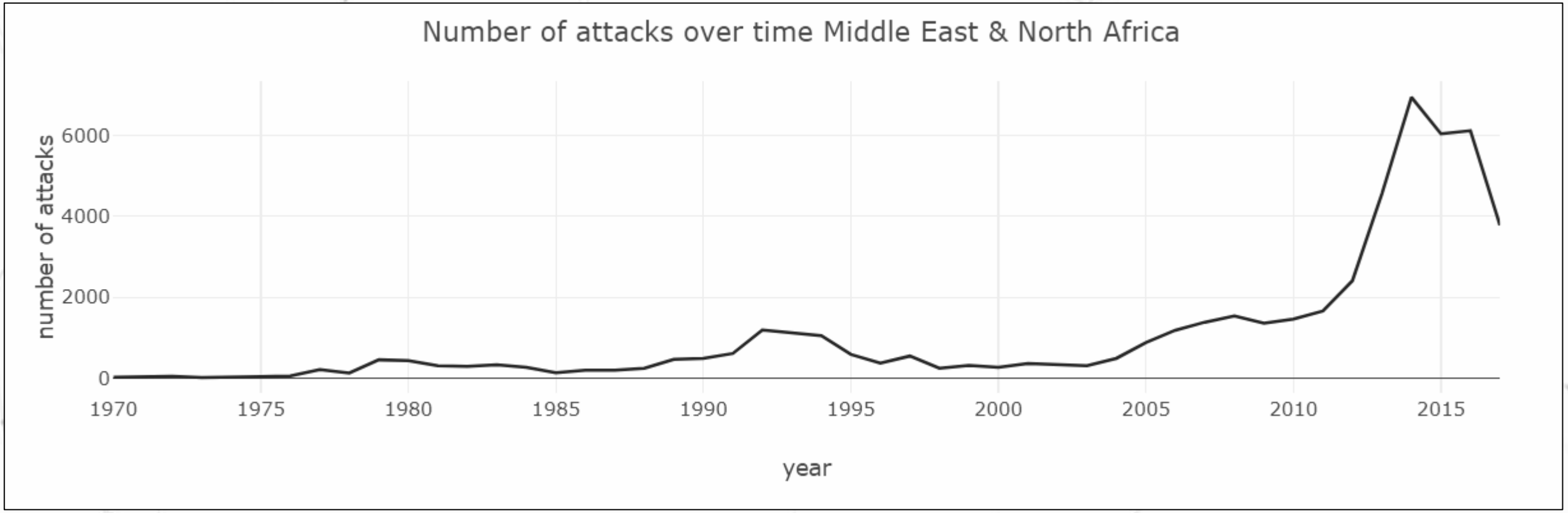
Within the Middle East, the majority of attacks are happening in Iraq…



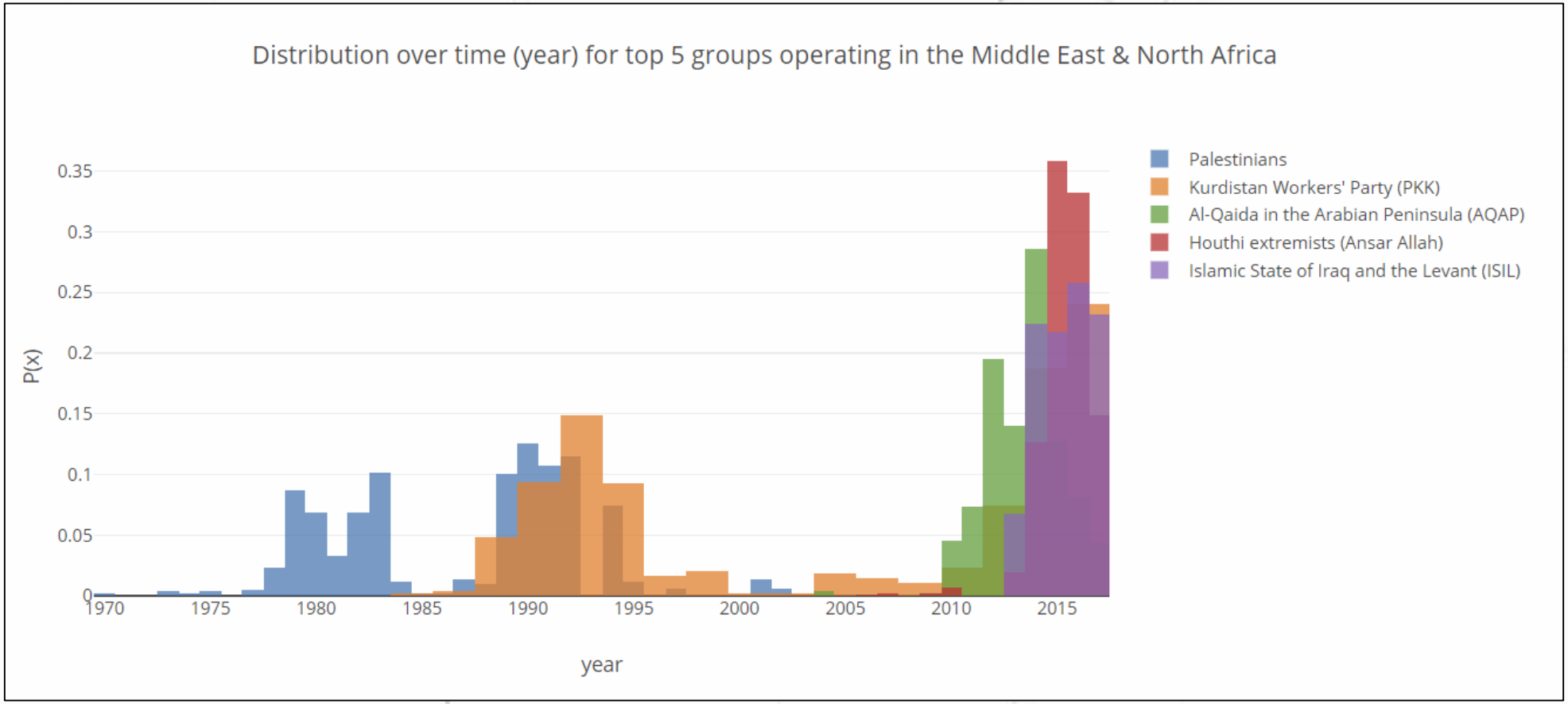
Though country location information seems highly relevant, with some groups operating exclusively in certain countries, suggesting country to be a good explanatory variable for differentiating groups…



If we look at the number of attacks in the Middle East as a time-series we see there has been a big jump in recent years again reinforcing the pertinence of focusing on this region



Though it seems that some groups have started (and/or stopped) operations within given time ranges, again suggesting time may help differentiate groups too…

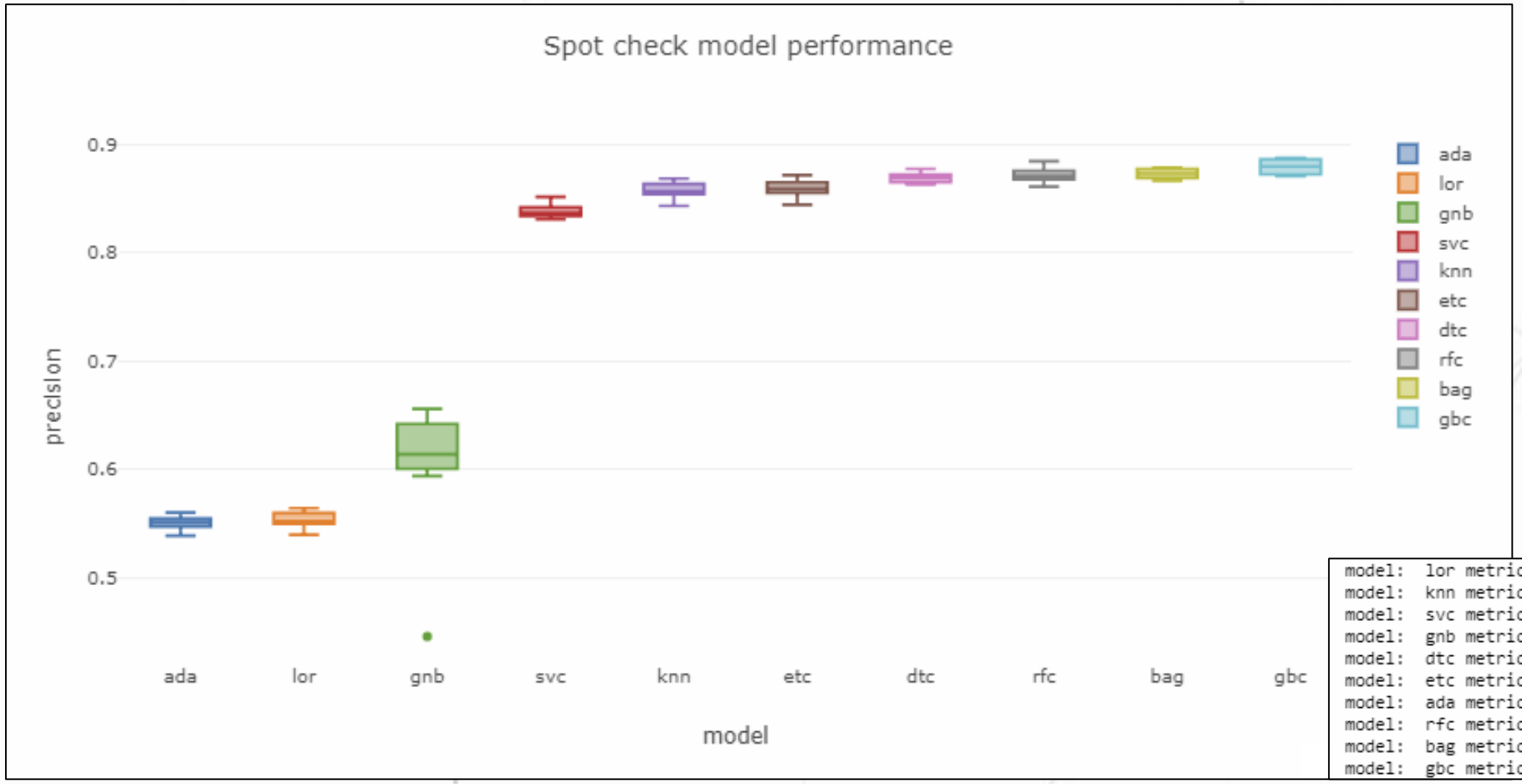


10 classification models were spot-checked in V1 using just five explanatory variables and no model tuning. This was to assess which type of models might be bested suited to this problem space and to set a baseline for improvement

V1 PIPELINE & MODELS

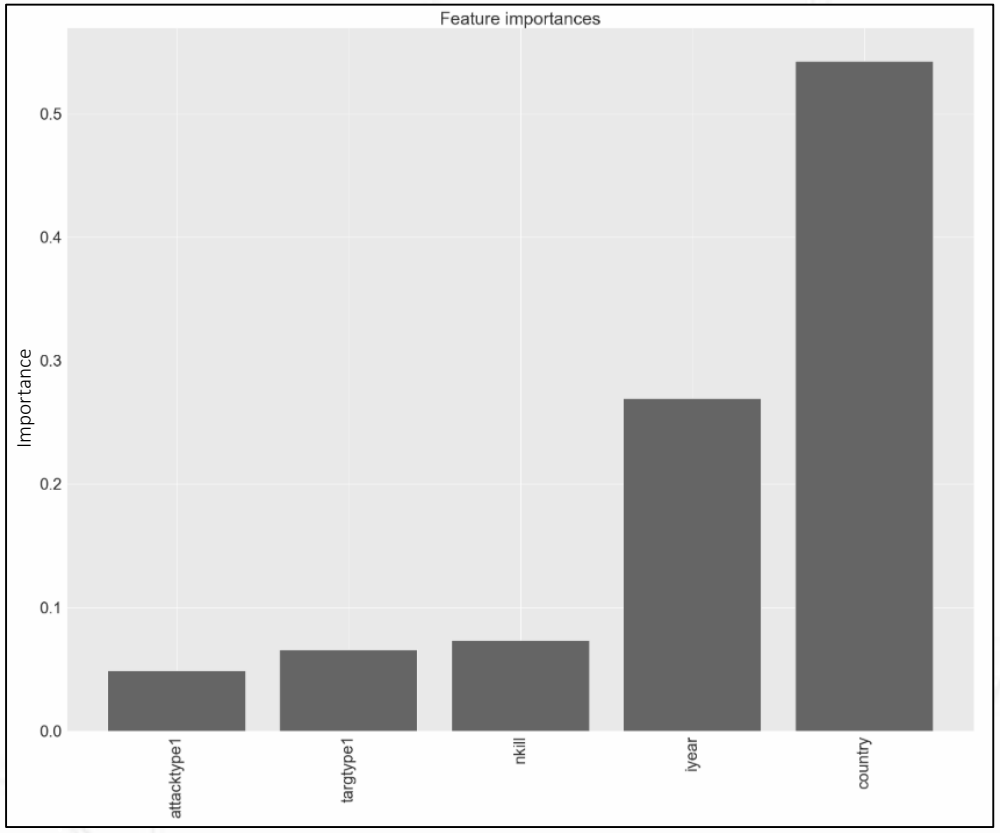
- Basic data-pipeline:
  - 5 variables, based on EDA analysis above (country, iyear, nkills, attacktype, targtype1)
  - Basic X-variable cleaning (conforming to correct data type, replace NAs with -99 codes)
  - Basic y-target variables cleaning (conforming to categorical, removing 'generic' groups)
- Basic machine learning evaluation functions:
  - K-fold cross validation for training/testing splits (10 splits, test = 20%)
  - Precision cost function (due to project goals, though acc/recall/f1/f1_weighted as options)
- Spot checking 10 different 'out-of-the-box' classification algorithms:
  - Linear models: Logistic Regression, Ridge Regression,
  - Non-linear models: K-Nearest Neighbours, Classification trees, Extra tree, SVM, Naïve Bayes
  - Ensemble models: AdaBoost, Bagged Decision Trees, Random Forest, Gradient Boosting Machines

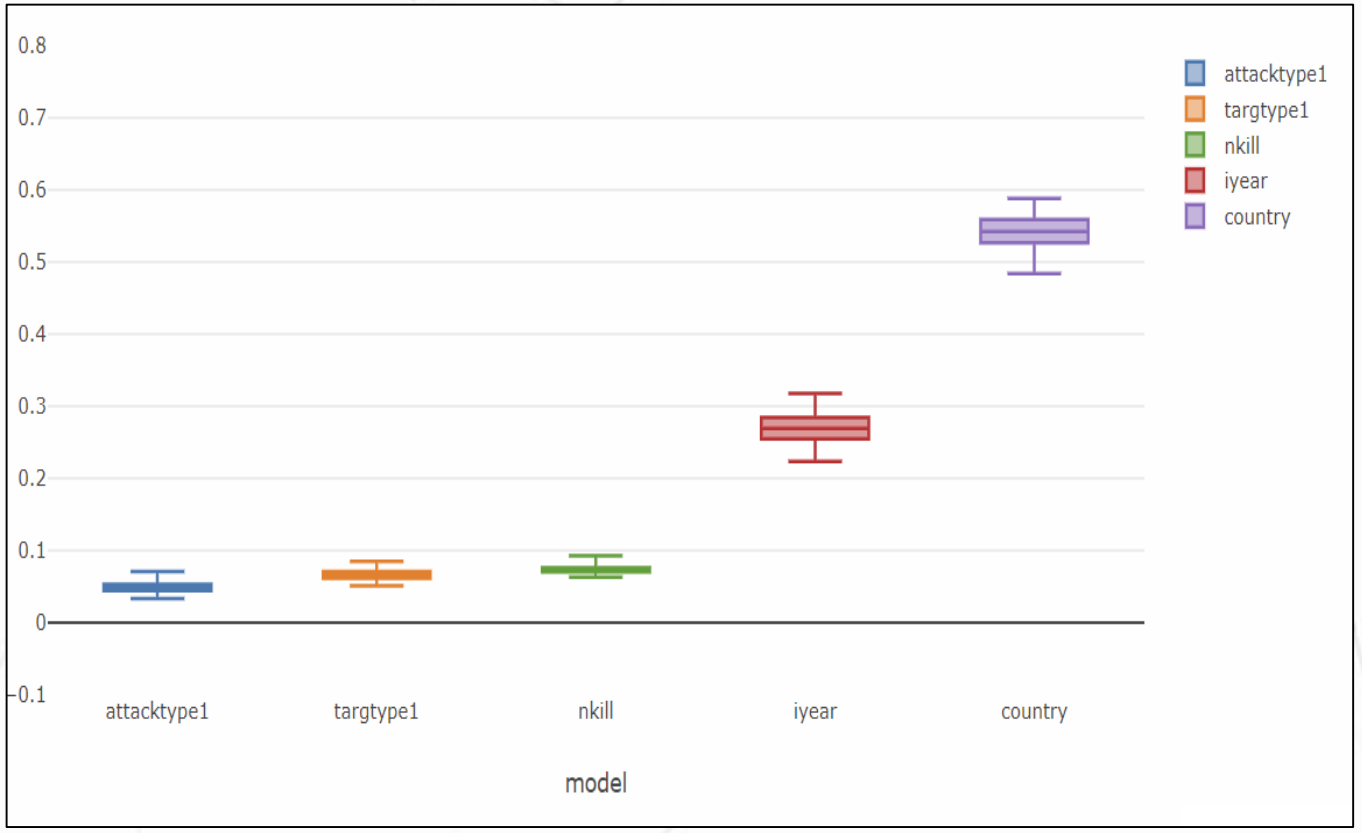
Ensemble models generally seemed to perform well (likely due to the non-linear nature of the problem, the unbalanced classes, and the categorical nature of most variables).



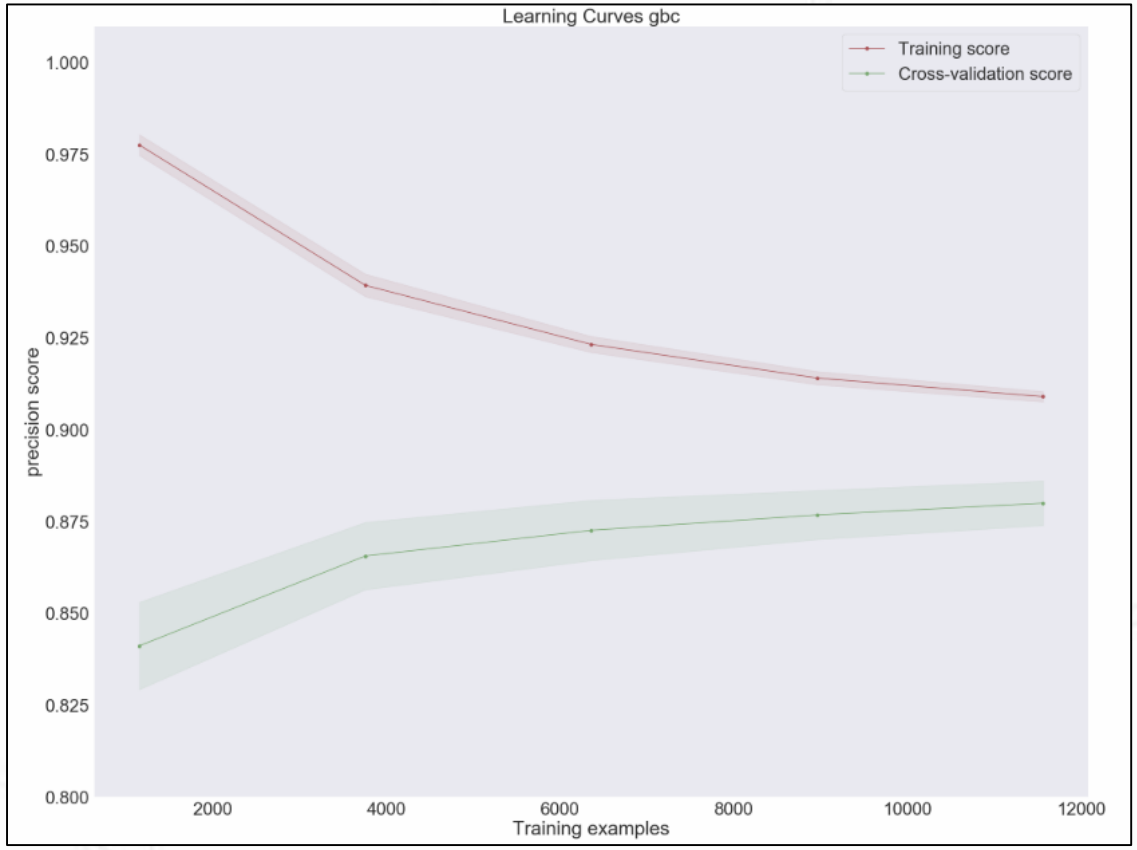

```
model:  lor metric, precision:  0.553 time:  12.70
model:  knn metric, precision:  0.858 time:  4.114
model:  svc metric, precision:  0.839 time:  247.8
model:  gnb metric, precision:  0.605 time:  0.696
model:  dtc metric, precision:  0.869 time:  0.715
model:  etc metric, precision:  0.859 time:  0.575
model:  ada metric, precision:  0.55 time:  18.753
model:  rfc metric, precision:  0.872 time:  2.640
model:  bag metric, precision:  0.873 time:  3.345
model:  gbc metric, precision:  0.88 time:  317.25
```

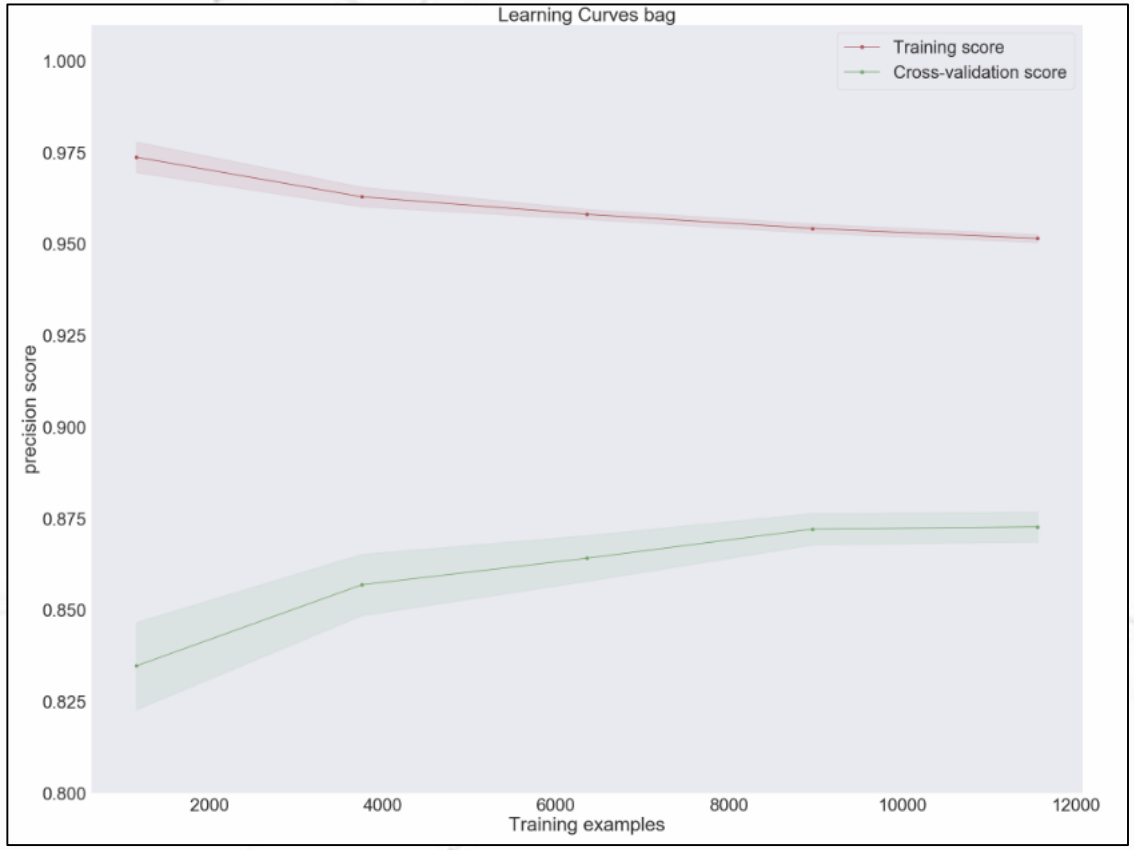
The strong performance of many V1 models suggests that some of the key explanatory variables are already contained, and looking at the (tree) feature importance it seems clear that this is largely from country and year as hypothesized:



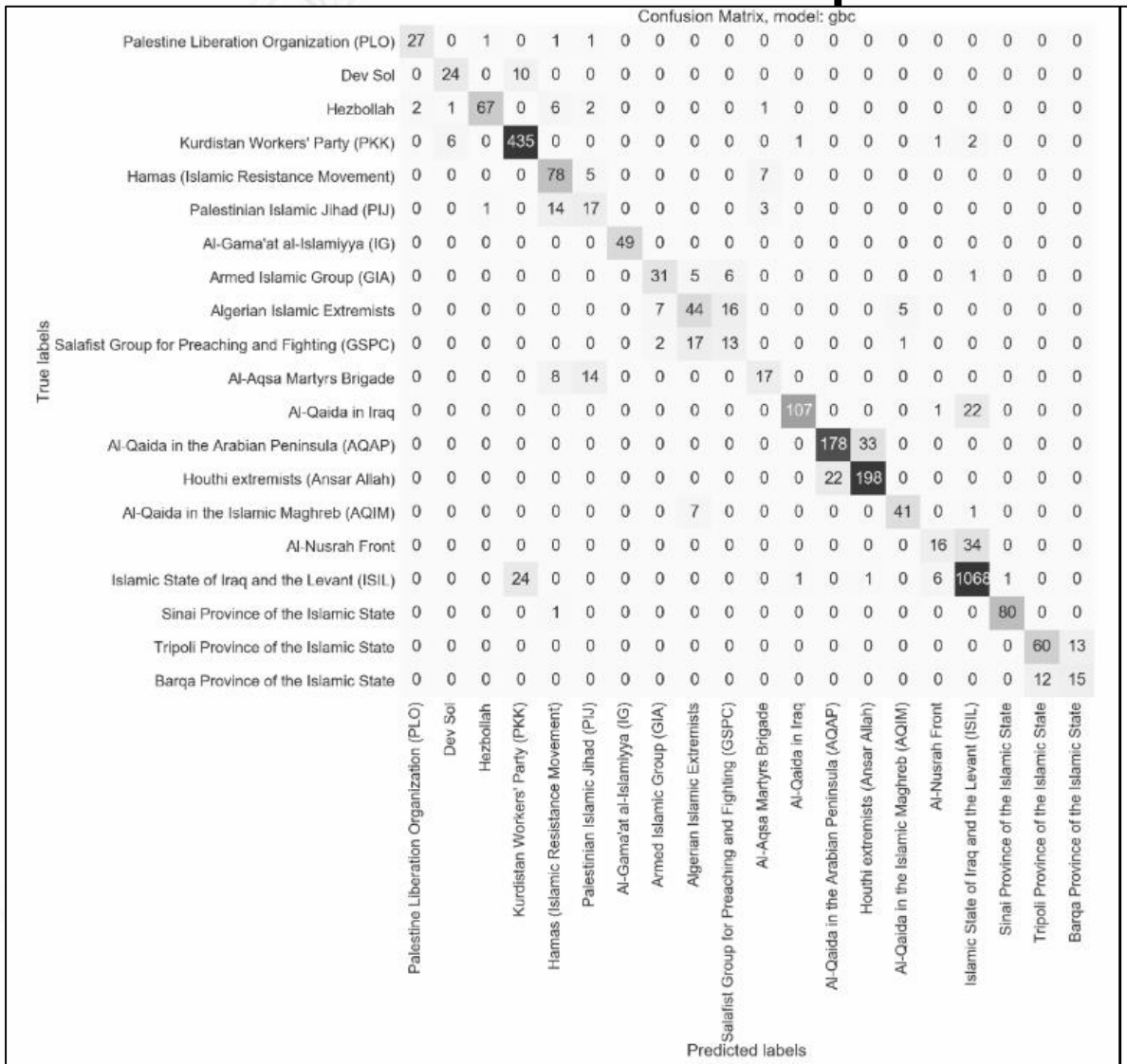


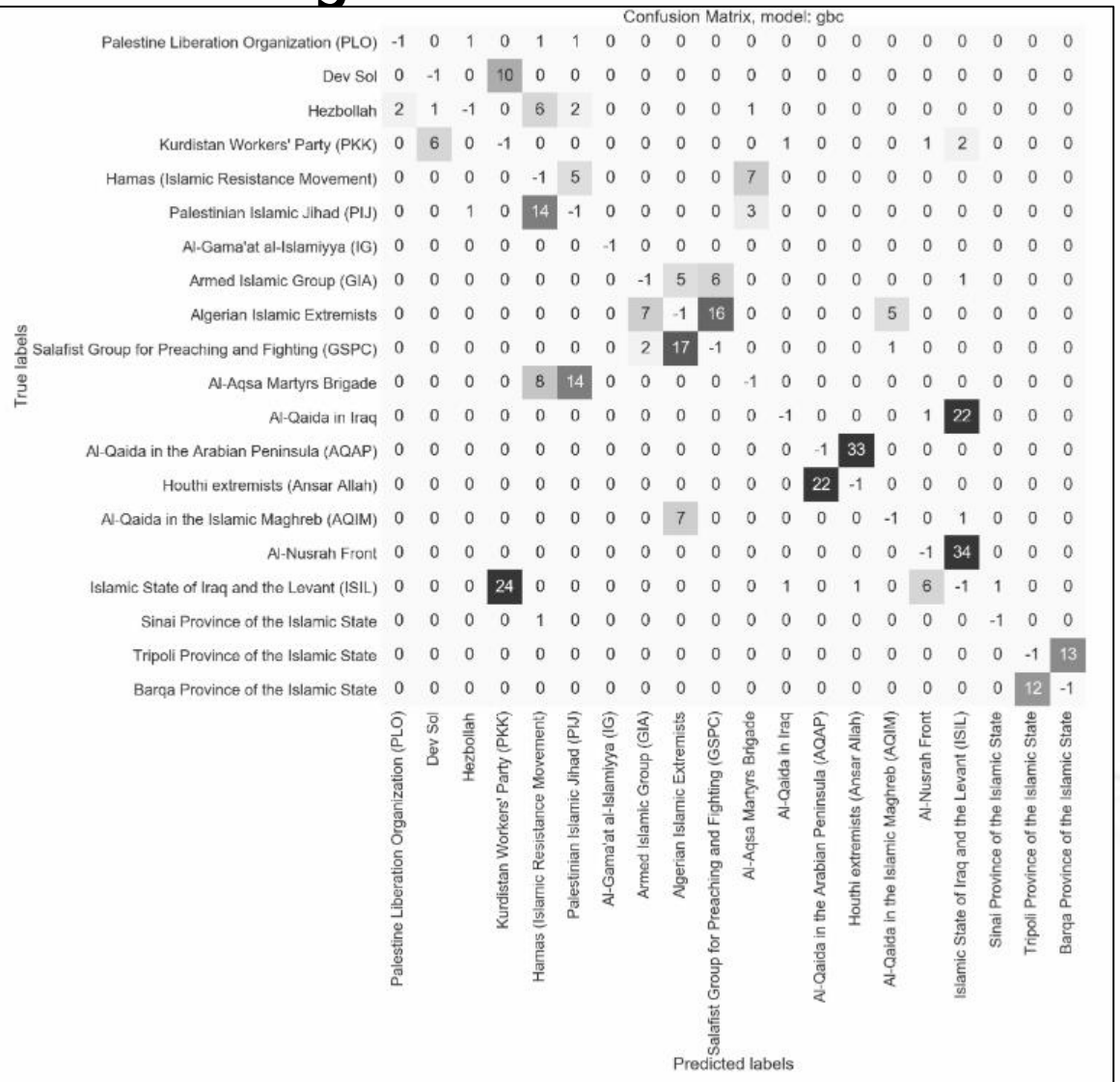
Looking at the learning curve for the top performing models (GBC, Bag), it seems the model was able to generalize well, though there are still concerns they may be overfitting





The confusion matrix evaluation for the top model of V1 is generally looking good with most predictions matching the ground truth of the test set, though it seems there is still clear room for improvement with some significant misclassifications…





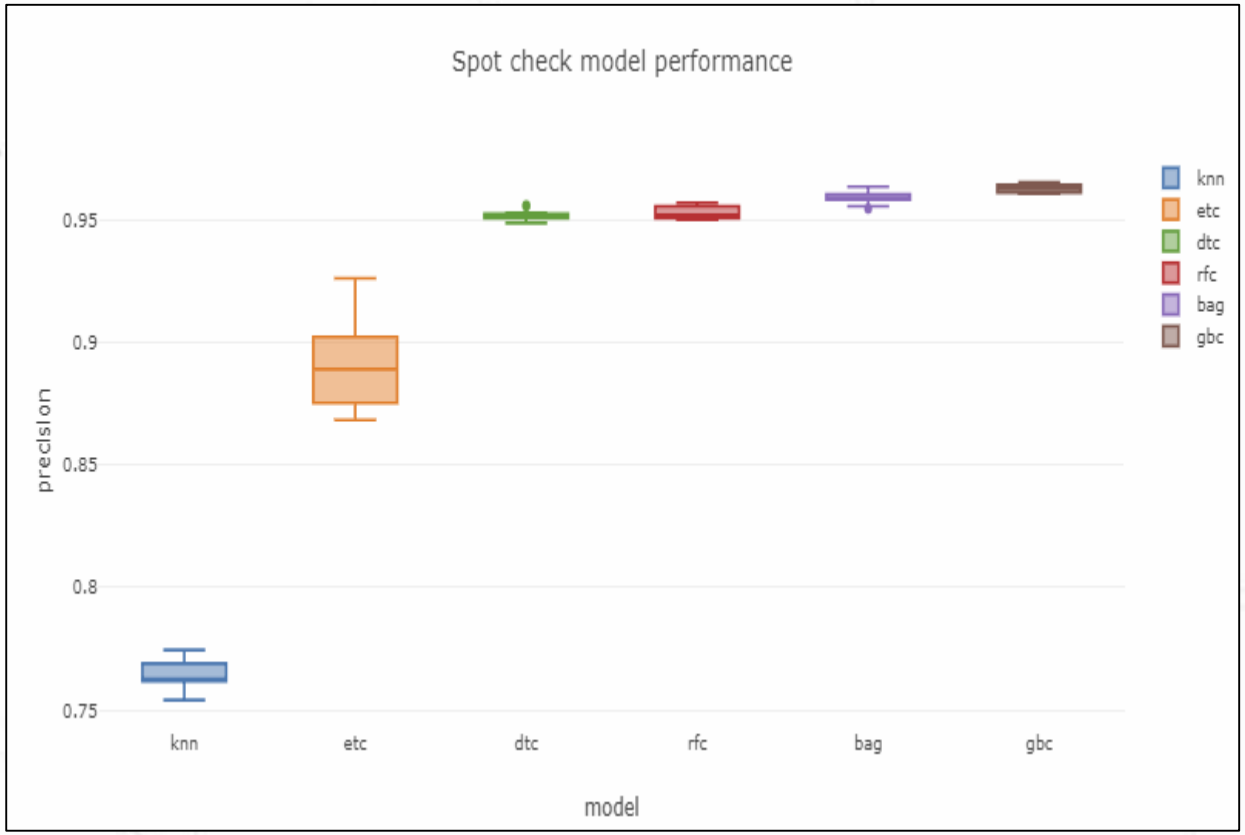
Building on the previous pipeline, V2 assessed over 40 other relevant features including field-specific cleaning and transformation to try and improve results

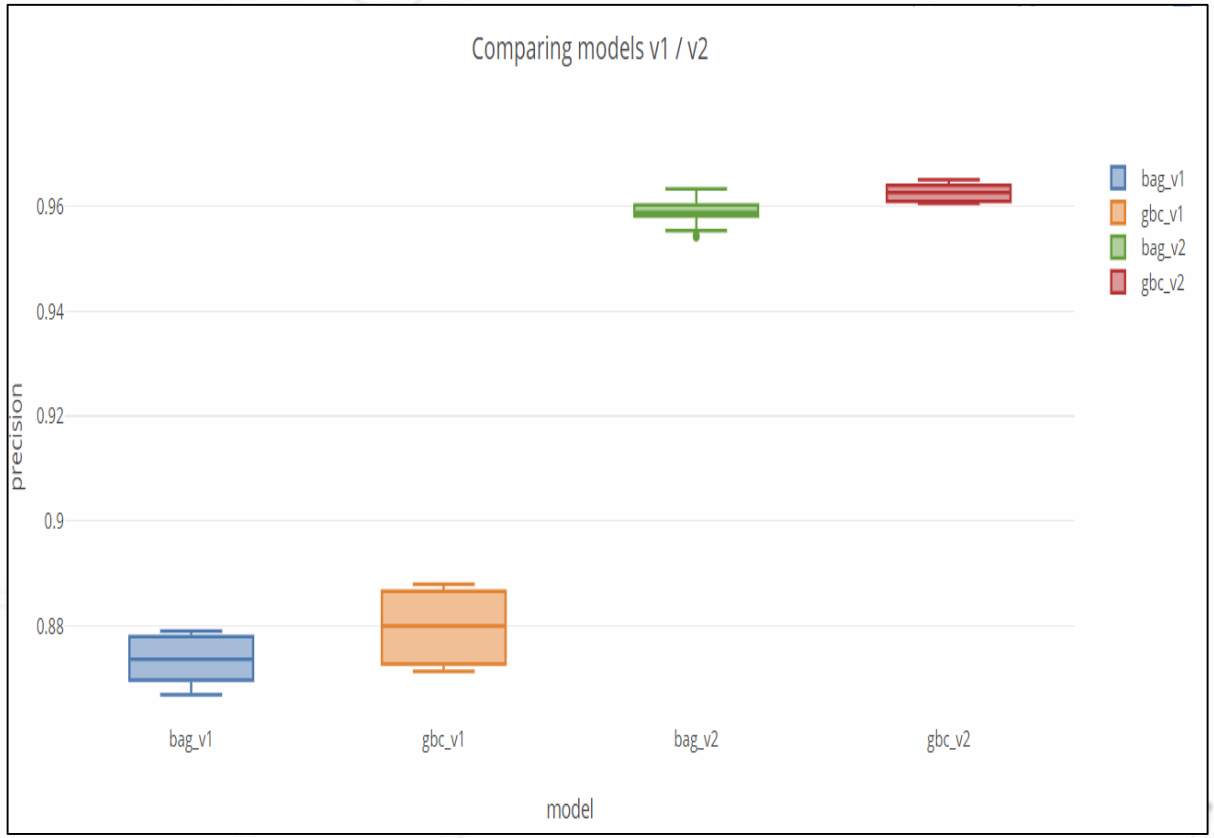
V2 PIPELINE & MODELS

- Basic data-pipeline:
  - 24 variables finally included (42 assessed), based on manual EDA of all fields in GTD codebook*
  - Deep X-variable cleaning (datatypes, replacing NAs with codes, normalisation etc.)
  - Basic y-target variables cleaning (conforming to categorical, removing 'generic' groups)
- Basic machine learning evaluation functions:
  - K-fold cross validation for training/testing splits (10 splits, test = 20%)
  - Precision cost function (though acc/recall/f1/f1_weighted as options)
  - Upsampling functionality was added to reduce class imbalance issues
- Spot checking 6 best perfoming algorithms from v1:
  - Non-linear models: K-Nearest Neighbours, Classification trees, Extra tree
  - Ensemble models: Bagged Decision Trees, Random Forest, Gradient Boosting Machines

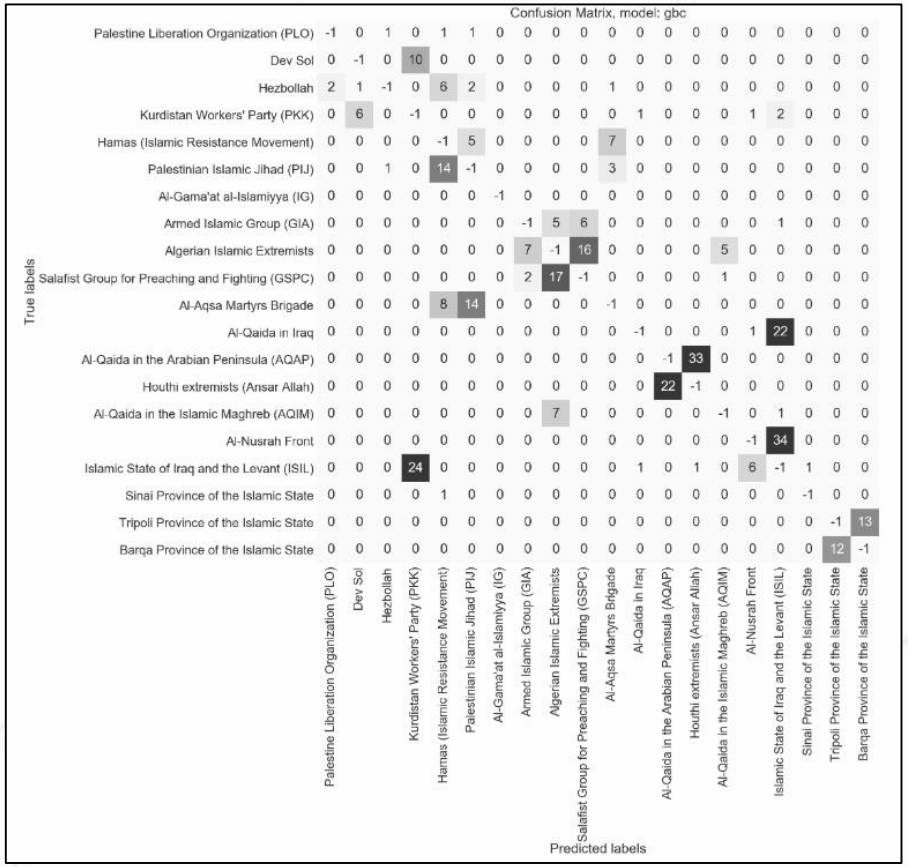
* https://www.start.umd.edu/gtd/downloads/Codebook.pdf

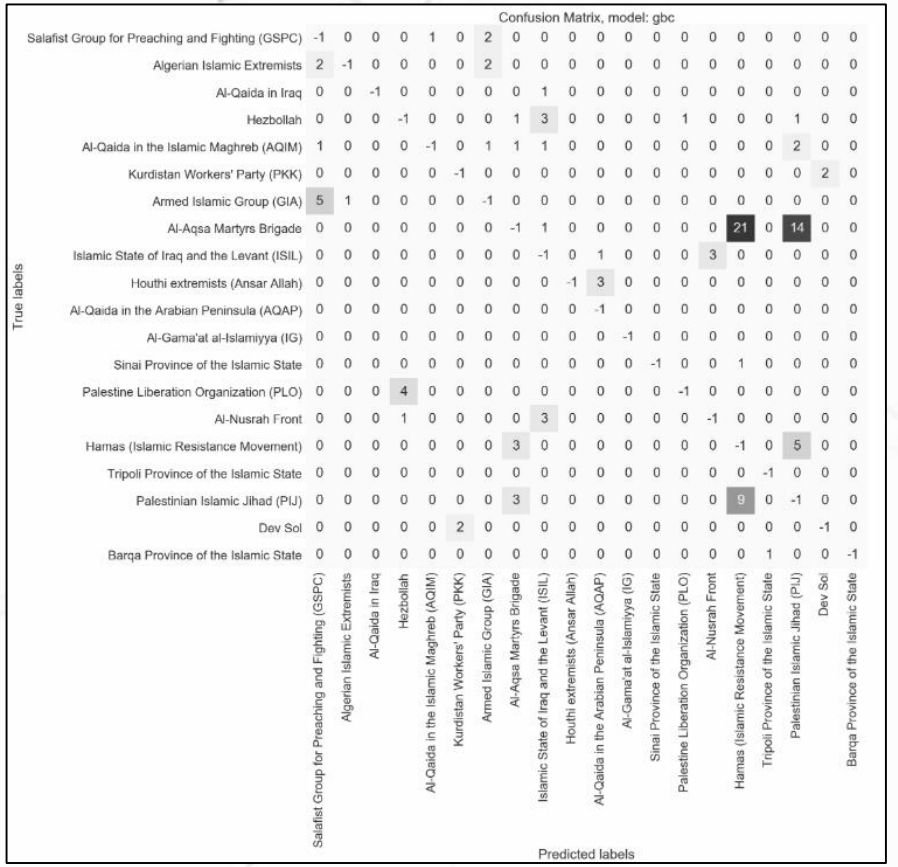
While all models perform significantly better with the additional (cleaned) features, Gradient Boosted and Bagging Classifiers were the best (though GBC was slowed down significantly by the new features in terms of run-time)



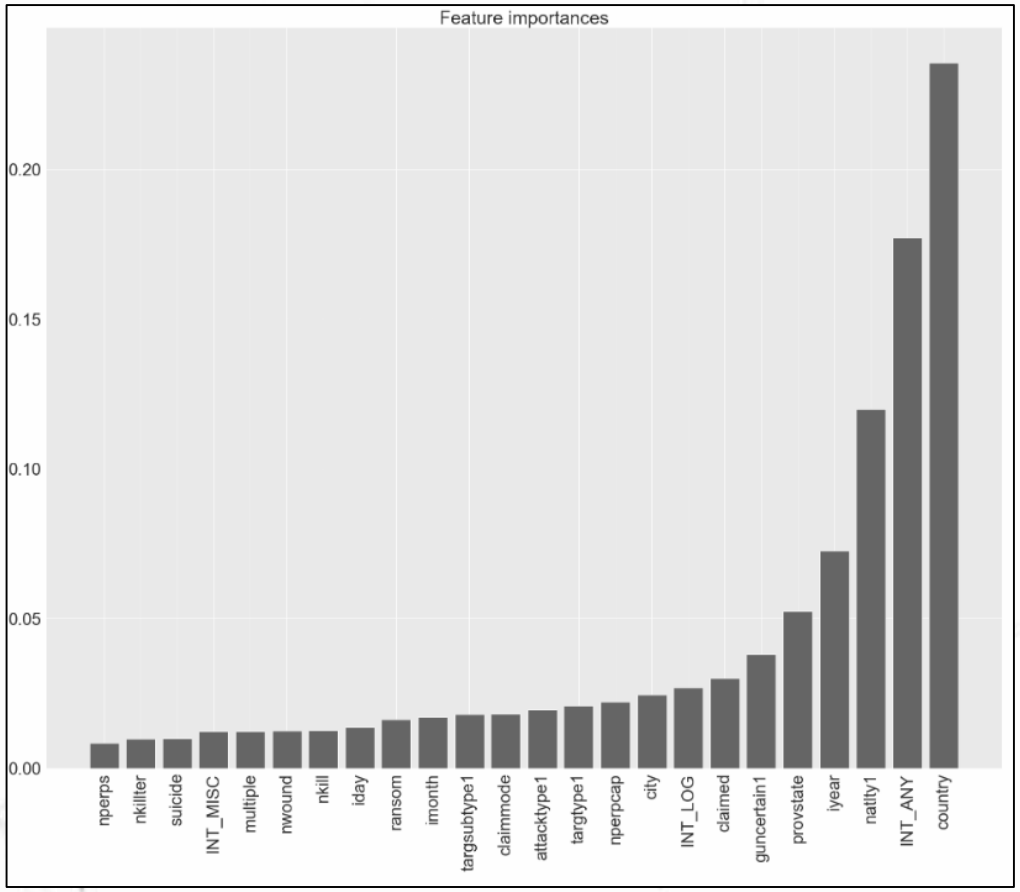


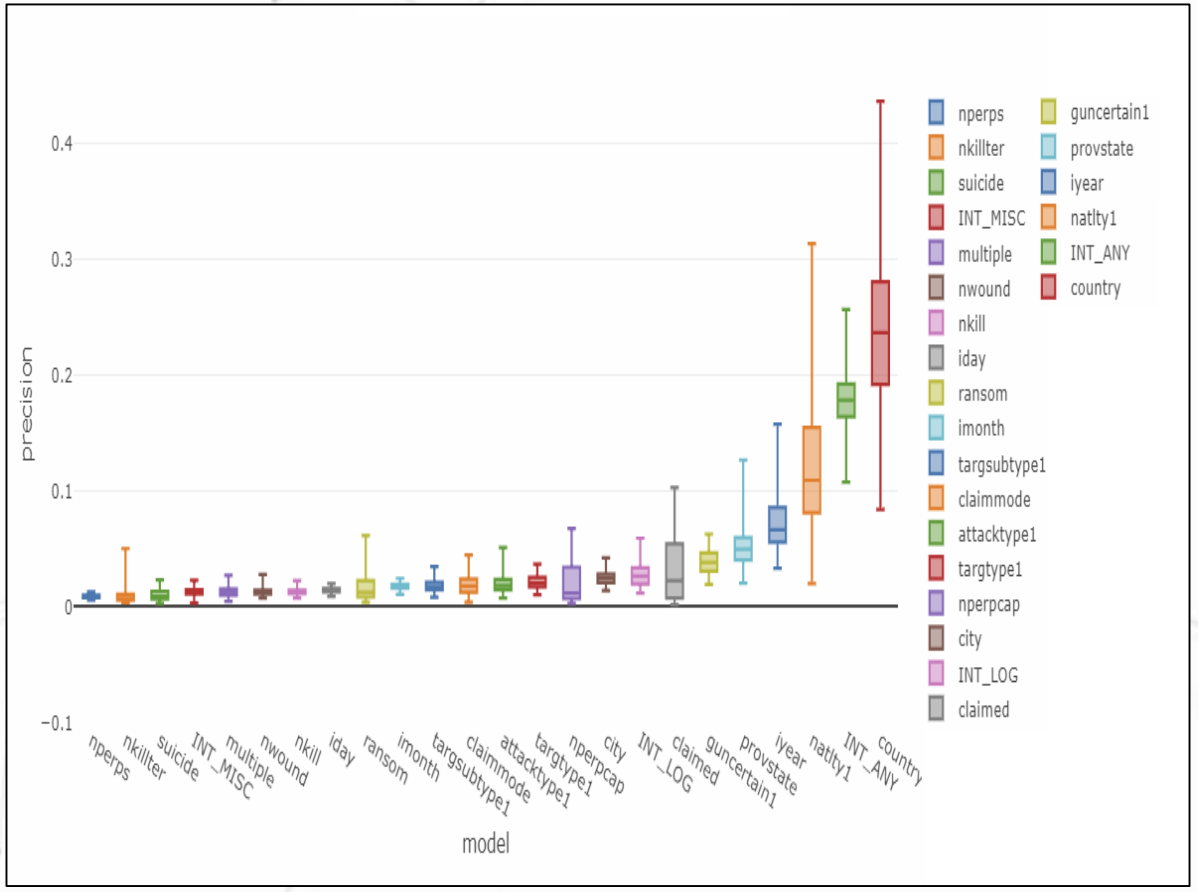
With the best performing model of v2 showing systematic improvements in the confusion matrix compared to v1…





And while many features added in v2 were highly significant (like the nationality of the victims, natlty1), they showed varying amounts of importance suggesting some could be dropped in the final model (e.g. number of perps)…





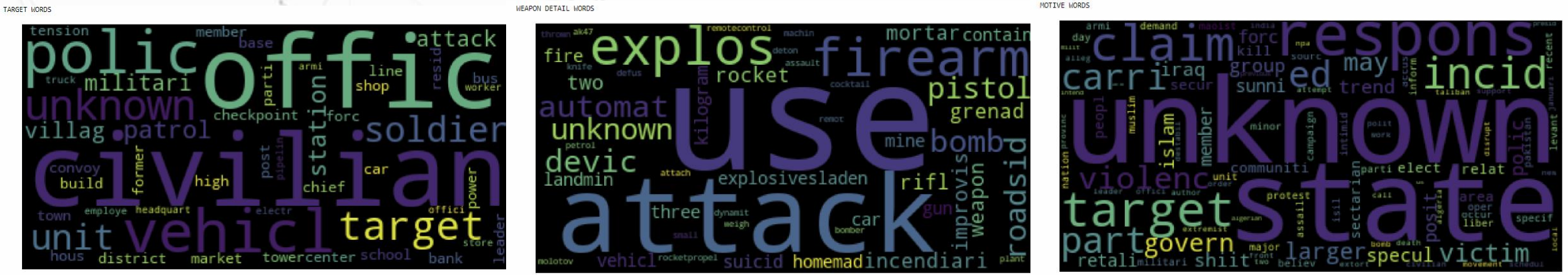
Building on the previous pipeline, V3 attempted to build new features by processing various text fields (e.g. 'motive') into machine readable features using NLP techniques

V3 PIPELINE & MODELS

- Basic data-pipeline:
  - 27 variables included, based on manual EDA of all relevant fields from GTD codebook*
  - 200 word vectors based on the motive (100), target1 (50), and weapdetail (50) text fields
  - Deep X-variable cleaning (datatypes, replacing NAs with codes, normalisation etc.)
  - Basic y-target variables cleaning (conforming to categorical, removing 'generic' groups)
- Basic machine learning evaluation functions:
  - K-fold cross validation for training/testing splits (10 splits, test = 20%)
  - Precision cost function (due to project goals, though acc/recall/f1/f1_weighted as options)
  - Upsampling functionality included
- Spot checking 2 best perfoming algorithms from v1:
  - Ensemble models: Bagged Decision Trees, Gradient Boosting Machines

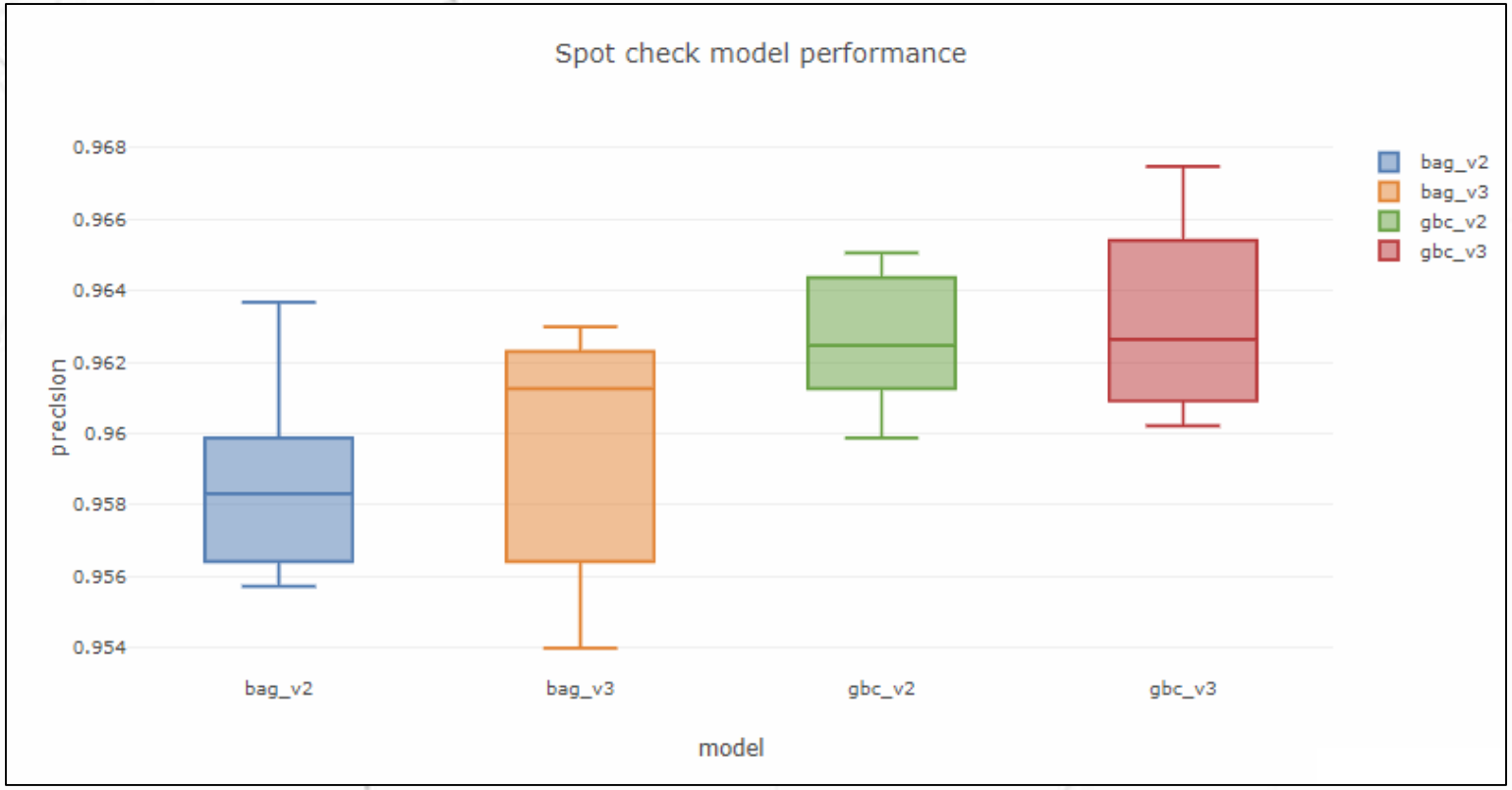
* https://www.start.umd.edu/gtd/downloads/Codebook.pdf

Using NLP techniques (tokenization, stop-word removal, stemming etc.) the top n words from text fields like ‘motive’ were transformed into vector fields in an attempt to draw out signal from potential patterns in group motives…



WordCloud of the top relevant words from the ‘target’, ‘weapon details’, and ‘motive’ text fields

While the NLP derived features did show some improvement, this was not statistically significant and served to increase run-time dramatically suggesting more work would be needed to improve the process if they were to be included in a future model…



Though the GBC model was marginally better overall than the Bagging classifier algorithm (of the models tests), there was an order of magnitude difference in run-time for almost negligible improvement meaning a choice would need to be made if run-time/hard-ware costs were more important than performance gains for the user of this model…

V2

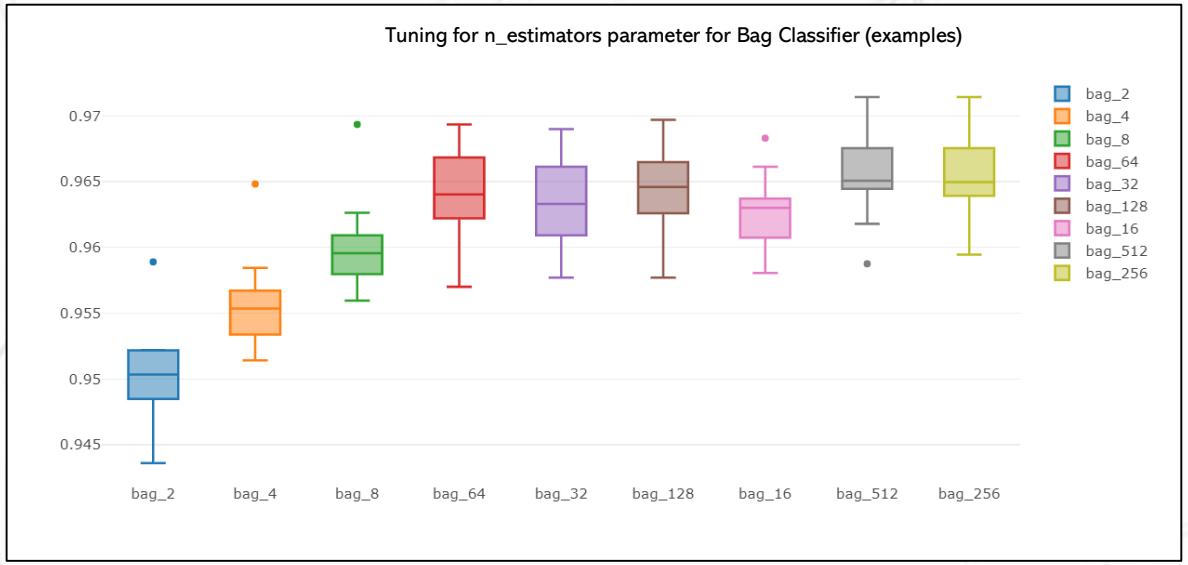
```
clean dataframe created:  shape -  (14452, 27) number of NAs -  0
model:  lor metric, precision:  0.803 time:  272.63627314567566
model:  knn metric, precision:  0.764 time:  15.79237151145935
model:  gnb metric, precision:  0.688 time:  2.5682926177978516
model:  dtc metric, precision:  0.952 time:  2.008671998977661
model:  etc metric, precision:  0.901 time:  1.2368690967559814
model:  ada metric, precision:  0.614 time:  24.043310403823853
model:  rfc metric, precision:  0.952 time:  3.90964412689209
model:  bag metric, precision:  0.959 time:  9.067947149276733
model:  gbc metric, precision:  0.963 time:  556.6787431240082
```

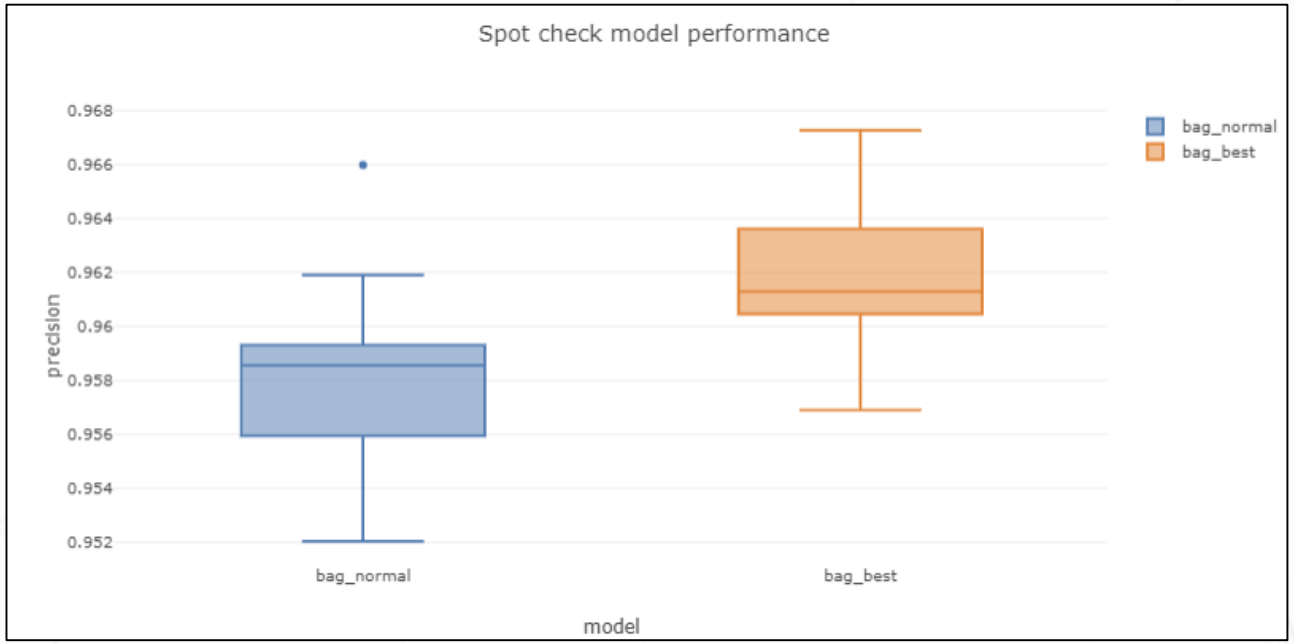
V3

```
clean dataframe created:  shape -  (14452, 227) number of NAs -  0
model:  bag metric, precision:  0.96 time:  51.121108531951904
model:  gbc metric, precision:  0.963 time:  2766.0724170207977
```

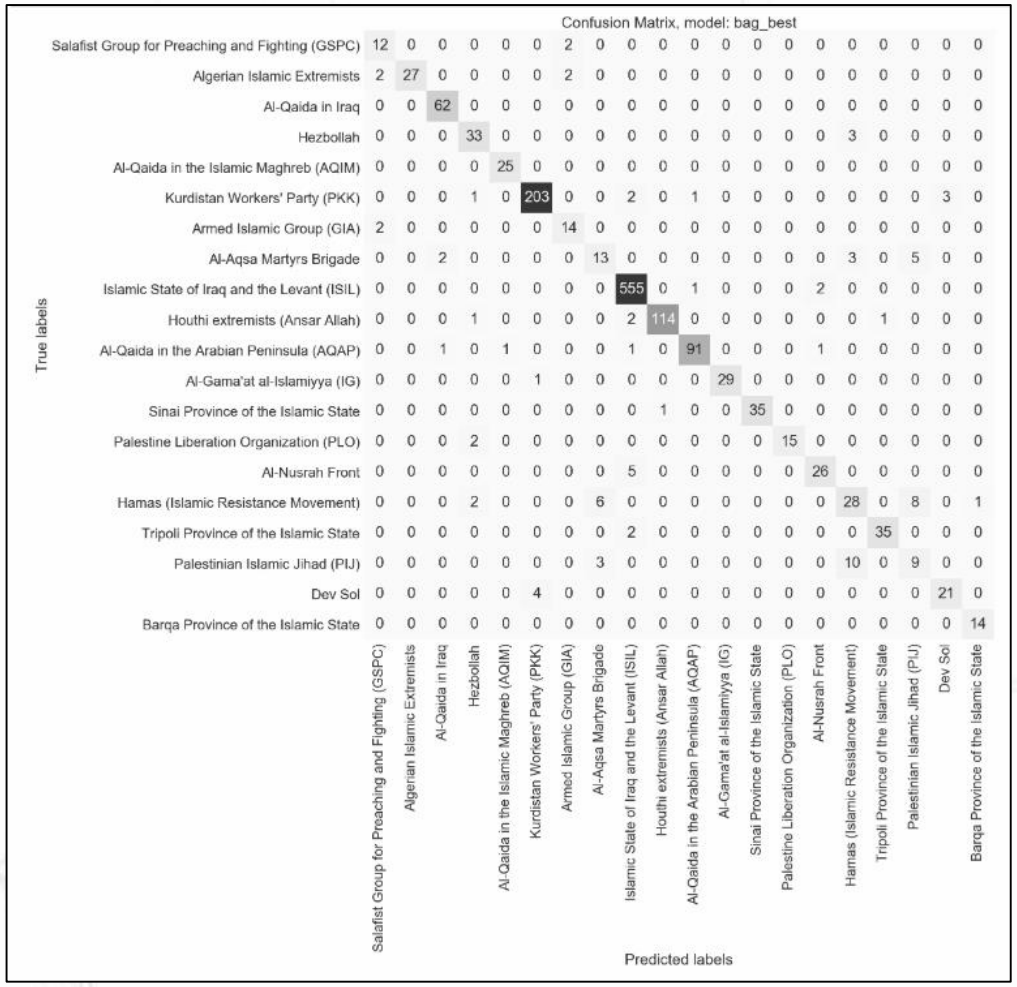
Ultimately Bagging Classifiers were selected as the final model because of the large run-time savings for essentially the same final performance

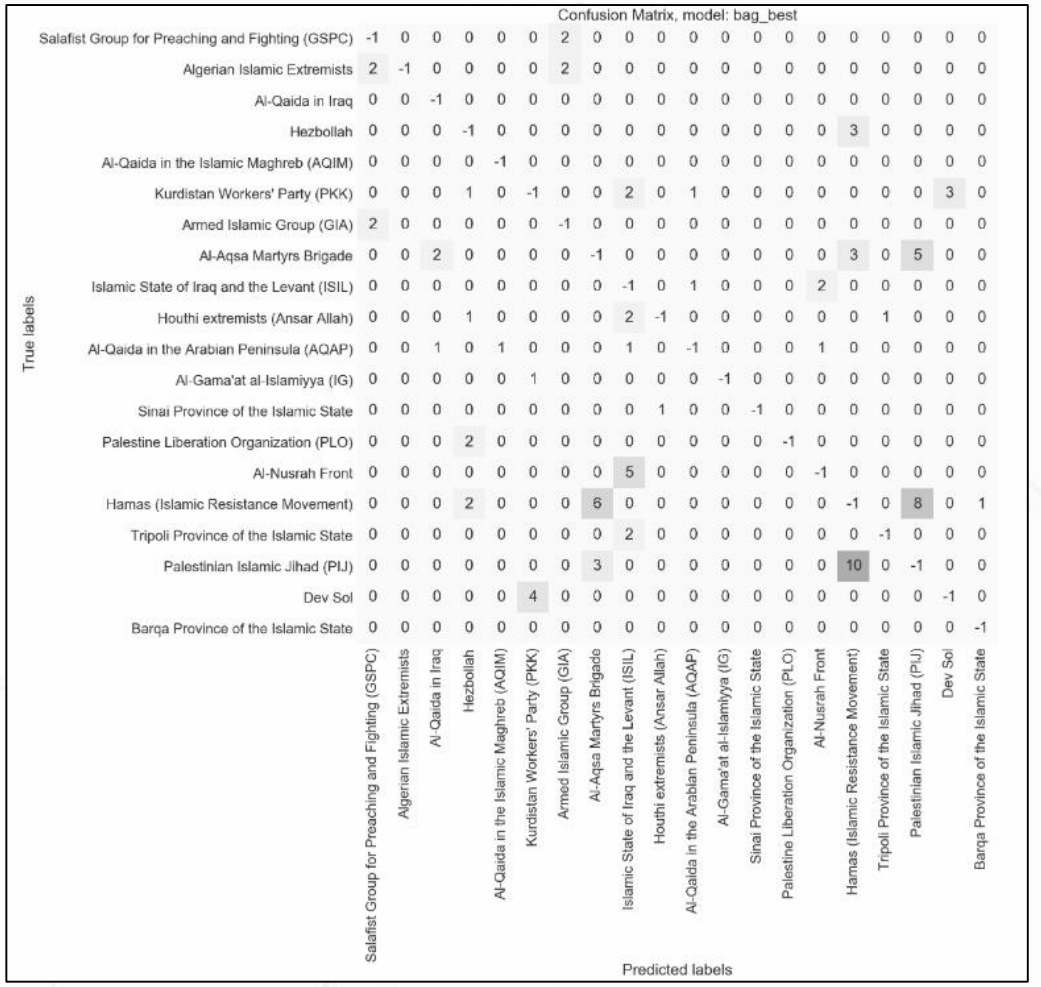
A final evaluation was therefore removed from the training/testing set and Grid Search was run over key hyperparameters to tune the final model, which showed significant improvements:





This final model was then redeployed on the evaluation set (i.e. a subset of the data never seen by the model before) on which it continued to perform well on, suggesting the model to be fairly robust!

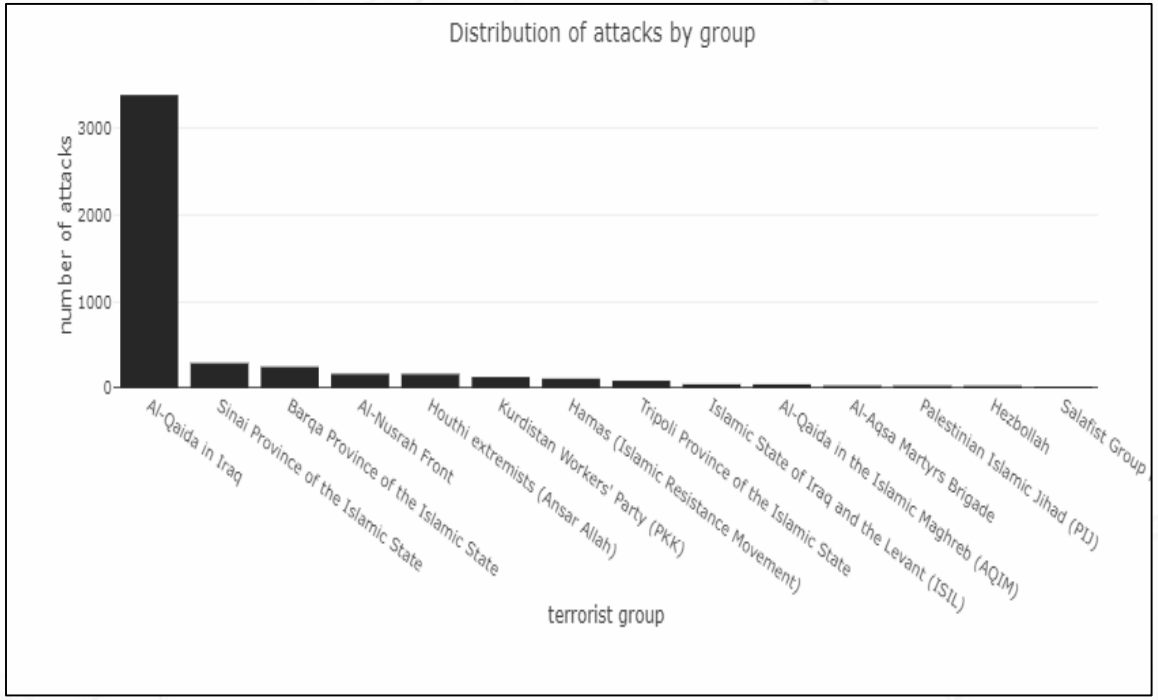


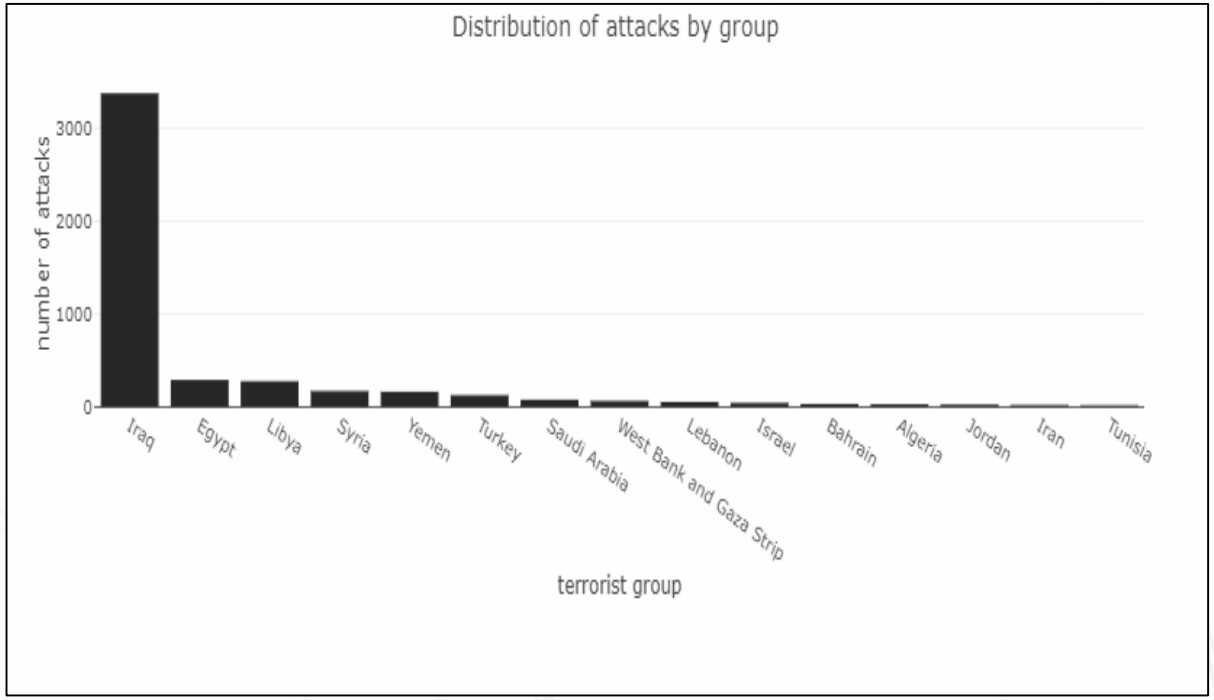


Whilst there are still many ways the current process could be improved, the last step is to deploy this final model onto the 'Unknown' attacks in the Middle East, both as an example of how it could be deployed in reality (as per our project goals), and to common-sense check the results to help understand if the model is performing as expected…

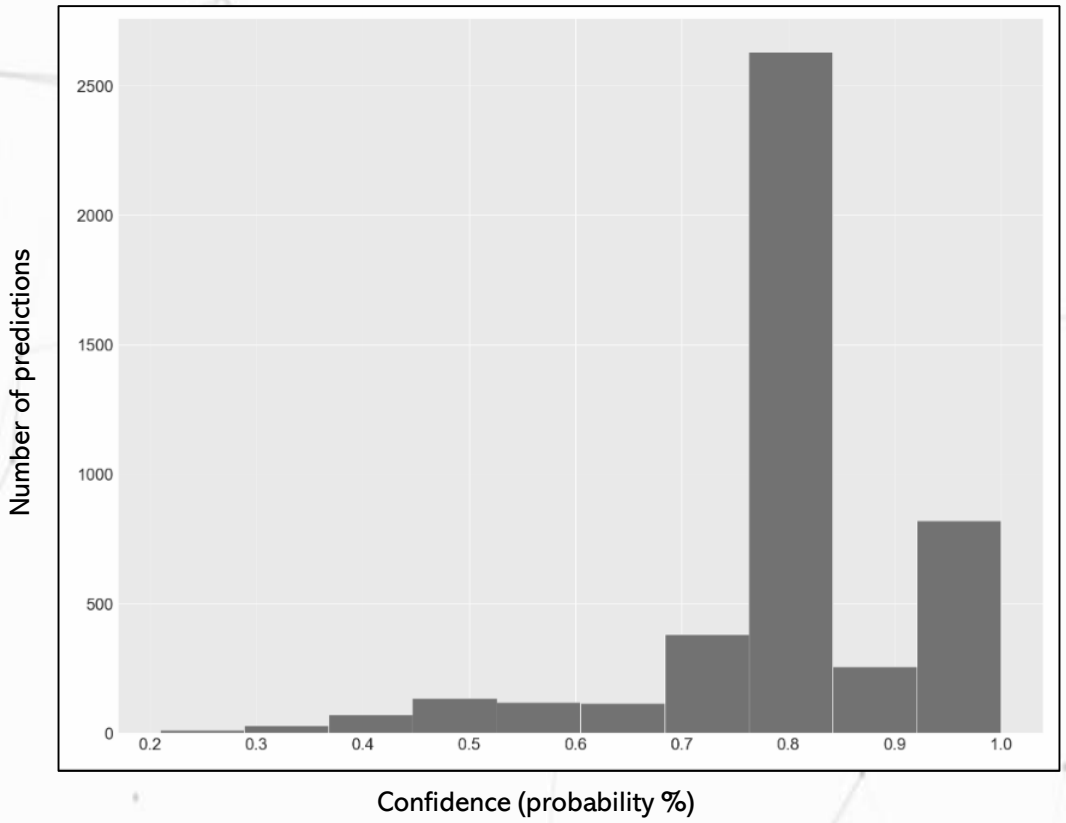
… to do this we took all Unknown attacks in the Middle East & North Africa which occurred in 2016 and 2017 (of which there were ~1700) and predicted the group responsible using our final trained model in an attempt to understand the culprits and help inform policy

While there is no way to objectively test if these predictions are correct (by definition) looking at the outputs it seems there are no obvious bugs (such as all predictions being in one class) and – whilst there are quite a large amount in one group – this is proportional to the unbalanced country location of attacks, suggesting, as hypothesized, that location is a major predictive variable





Looking at the confidence level of predictions, then generally we find the model is quite confident in most cases, though to deploy this model in form policy we would need to define a threshold in line with the acceptable limit to help make policy decisions (e.g. >80%), in which case we would cut the threshold here



A number of key insights can be drawn out of this analysis…

1. It is important to set **clear real-world project goals** in order to help define and focus down the technical aims of analysis
2. The strong link between group activity and locationality means **geographic variables are very predictive** and there is a good reason to make regionally specific models
3. Given regional specificity it made sense for this analysis to **focus on Middle East and North Africa** due to a large number of attacks in the area suggesting the project may have most impact here
4. Given the disproportionate impact a small number of active groups have it made sense to also **focus analysis onto these most active groups**
5. For groups operating in the same geographic areas, **cleaning and selecting other variables** like nationality of victims attacked and method of attack **proved significant discriminators**
6. **Non-linear ensemble models worked well** for this data set, likely due to their ability to deal well with unbalanced classes, the large number of possible variables, and the fact that many of these were categorical variables
7. Using **precision as the cost function** for this problem makes sense as we want to be confident we are right if we are going to assign responsibility to a group to make potentially deadly policy decisions
8. Using NLP to create new features from **text info on the attacks was not able to significantly improve results**, though more work may be able to draw more signal out of this
9. **Tuning model parameters was able to improve performance** for the best performing ensemble method, with results standing up well even to hidden evaluation sets the model has never seen before
10. This model could be **deployed to help attribute responsibility for recent and new unknown attacks** in the Middle East and could thus **feed evidence for policy decisions in the area**, such as which groups to target

Though it is important to also be self-reflexive of where some weaknesses remain

1. While **only looking at the top groups** provided a good way to reduce down the problem space while still capturing the highest value targets, this approach does present two problems. Firstly it means the currently model is unable to capture the long-tail of other smaller outfits which may be salient to understand from a policy perspective (i.e. being able to identify and shut-down groups which are increasingly violent early may be beneficial from a policy angle). Moreover only giving the model the target space of these top groups will mean it is only able to classify attacks as being carried out by one of these groups, potentially meaning that salient patterns from other smaller groups are always hidden by this model's outputs. This could be improved by extending the analysis out to all known groups, though this could come with new issues such as a greater need for balancing classes etc.
2. There is still some concern given the non-linear nature of the final models and the relatively small amount of data (meaning only a small evaluation set could be used) that the **model is still over fitting to some extent**. This could be improved by testing the model across further evaluation sets (i.e. having a larger K for the cross-validation of the last step), as well as double checking them are no bugs meaning that data from training sets is being used in evaluation.
3. If more time was available more effort could also be put into **tuning the final model** with only a small set of hyperparameters for the single best performing model used. This could be improved by doing a more exhaustive grid search of the final model as well as other top performing models to ensure optimal tuning. The best strategy to do this would be to host the source code on an (AWS EC2) instance and allow the search to run on a dedicated server.
4. While precision made sense to use in this problem, it may also be **worth revisiting the cost function** and using something more sophisticated like weighted F1 with a beta-score balanced between recall and precision depending on real-world priorities. This would require further refining of the project goals.
5. **Terrorism is a complex topic and evolves over time** (e.g. new groups could emerge, old groups might change leadership and faction off into different distinct groups, or new lone-wolf actors could emerge acting differently from the group they are supposed to represent). This suggest that we would need to get new data from the GTD and review their definitions regularly, as well as retraining the model itself often
6. Linked to the point above, it should also be noted that there is a good likelihood, considering the domain space, that parts of the **data itself are labelled wrong**. If this model was put in deployment the inputs and results would need to be assessed carefully to guard against issues this could cause.

Finally there a number of ways in which this analysis could be expanded and improved if there was more time:

1. **Extend to other regions:** As this analysis only covered the Middle East, it would be interesting and potentially valuable to extend this framework out to additional regions, functions have been built with regionally specific inputs to make this easier.
2. **Extend to long-tail of smaller groups:** As previously noted, extending the model to cover the long-tail of smaller groups would improve the real-world value of this model by helping identify anomaly attacks as well as trends in smaller but growing groups
3. **Country specific models:** Considering the high relevance of country as a predictive factor it may be fruitful to refine the model down even further to tune it on a per-country basis
4. **Advanced ensemble modelling:** With the clear efficacy of ensemble and gradient boosting models in the problem, it would be valuable to extend the modelling out to novel extreme gradient boosted models such as XGBoost which could have significant performance gains
5. **Better NLP processing:** While attempts were made to feature engineer useful variables from text fields, more work could be done to try and push this idea further to see if certain key words (or groups of words) related to terror groups motivations or related news on the attack could help to identify the group responsible
6. **Augmenting novel data:** Related to the above point, building a data pipeline to connect these attacks to related news and social media data may help further draw out signals using NLP to determine group responsibility

Thank you

samuel.king@skanalytic.com