

Final Project: Terrorism Risk Prediction

18th June 2020

1. Team Members:

- Ariel Holin
- Dor Sklar
- Or Gindes
- Tomer Porat

2. OVERVIEW

- 2.1. Goal - Terrorism risk prediction for law enforcement organizations is an evaluation tool for prevention prioritization, based on intelligence gatherable prior to the event. The model was trained in the project using information regarding over 180k terror attacks spanning over 40 years.
- 2.2. Achievements - The model has achieved approximately 92.8% accuracy when predicting success or failure of an attack on previously unseen events, as well as similar precision with near perfect recall - meaning very few potential risky events will be classified as failures. The model also supports a decision breakdown, detailing what specific parameters have led to a specific event being considered a great or minor risk.

3. Final Results

3.1. Summary -

- 3.1.1. The final model was achieved using a process of NLP preprocessing on the 'Motive' text feature which resulted in word-features followed by dimensionality reduction (PCA) so that only the most influential word features were used (detailed in section 3.2). These features were added on top of the 'classic' features from the dataset and finally an XGBoost model, tuned by a GridSearch algorithm, yielded the results in Figure 1.

Accuracy Test: 0.926					
Accuracy Train: 0.935					
	precision	recall	f1-score	support	
0	0.69	0.19	0.29	832	
1	0.93	0.99	0.96	9281	
accuracy			0.93	10113	
macro avg	0.81	0.59	0.63	10113	
weighted avg	0.91	0.93	0.91	10113	

Figure 1 - Classification Report - where '0' label being attack failure and '1' label attack success

- 3.1.2. Due to the unbalanced nature of the dataset (89% of terror attacks are considered successful - meaning that attack was not prevented, regardless of whether the attack has achieved its goal), building an overly conservative model would be all too easy - simply predict majority class - all attacks being successful and have perfect recall.
- 3.1.3. The actual goal was to build a model with the best overall prediction accuracy and therefore, the most accurate feature importance estimation. To achieve this goal, the most useful metric to optimize was the `f1_score`, which is more informative than accuracy for a dataset as imbalanced as this one. This allows us to keep the model's recall high (not missing potentially successful attacks), while also striving for high precision, so that most attacks which the model considers a risk will actually be the attacks we should prioritize preventing.
- 3.1.4. With `f1_score` of 96% for predicting an attack's success, we can be reasonably certain that the model's probability estimations will be useful in prioritizing terror attacks prevention according to the risk they pose and that the decision making process of the model is sound. This result was only achieved using a boosting ensemble method (XGBoost) due to its advantage in predicting unbalanced distributions, where additional weight is afforded in each iteration to misclassified examples from previous ones.
- 3.1.5. The ROC curve metric (top of Figure 2) is quite useful when estimating the performance of this model, even though in this specific case, FP errors are not as important as FN errors (a failed attack being classified as successful isn't as bad as a successful attack being classified as a failure). It is useful because it can empower us to choose a decision threshold where we can maximise recall while keeping FP errors to a minimum and thus achieve the goal of the model - resource optimization via attack prioritization.
- 3.1.6. The PR curve on the other hand (bottom of Figure 2) isn't as useful in our case to estimate model performance, due to the imbalance in the dataset which means even a 'dummy' classifier would have seemingly high precision (Figure 3).

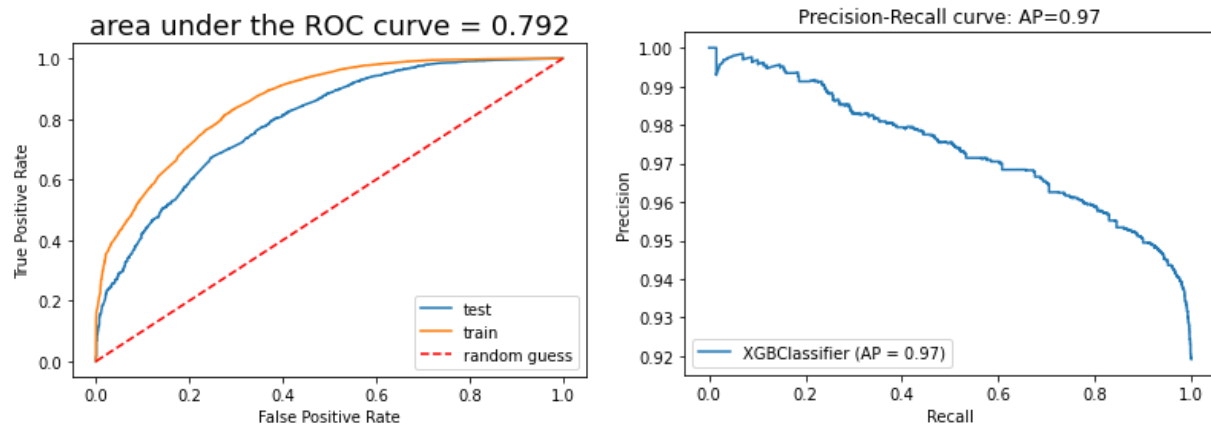


Figure 2 - ROC curve (left) and PR curve (right) for final model

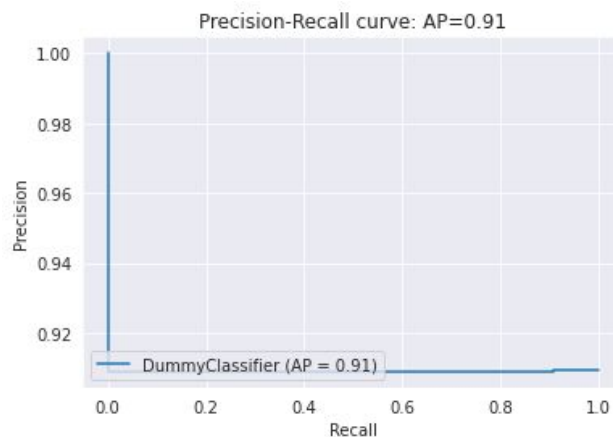


Figure 3 - PR curve for a random classifier

3.2. The following paragraph will expand upon the preceding summary with a detailed exploration of the relevant steps and process which led to the summarized results, as well as other techniques which were attempted in the process:

3.2.1. The data - A detailed and exhaustive EDA process was performed to gather intuition on the data and narrow down the relevant features we chose to use.

With consideration to possible applications of the model in mind, we chose to disregard features that could only be known after the fact (such as victims or property damage) and which might be too informative on the target variable and lead to leakage. Initially the features we used for the model were - Extended (event duration), vicinity (within or without city limits), 'Attack_Type, Weapon_type, Target_type, whether the attack was international and whether the attack was a suicide attack.

We considered taking 'Day' and 'Month' into account as well, but a Time-series Analysis, performed as part of the EDA, has shown no seasonality or time dependency, and so these features were dropped.

The dataset was split for train and test and the remaining features were treated and preprocessed to combine minor categorical values into the ‘Other’ category and both label encoding and one-hot encoding were attempted. The different methods of transformation didn’t have an impact on model metrics but they did provide different visualizations for feature importance so one-hot encoding was chosen so that the correlation of specific categorical values can be studied (Figure 4).

	var_1	var_2	value
0	Weapon_type_Melee	Attack_Type_Unarmed Assault	0.295094
1	Attack_Type_Bombing/Explosion	Attack_Type_Assassination	-0.360231
2	Weapon_type_Other	Attack_Type_Unarmed Assault	0.428431
3	Weapon_type_Explosives	Attack_Type_Armed Assault	-0.593715
4	Attack_Type_Bombing/Explosion	Attack_Type_Armed Assault	-0.614164
5	Weapon_type_Firearms	Attack_Type_Armed Assault	0.625267
6	Weapon_type_Firearms	Attack_Type_Bombing/Explosion	-0.754981
7	Weapon_type_Incendiary	Attack_Type_Facility/Infrastructure Attack	0.781214
8	Weapon_type_Firearms	Weapon_type_Explosives	-0.820231
9	Weapon_type_Explosives	Attack_Type_Bombing/Explosion	0.918342

Figure 4 - Most highly correlated categorical features

3.2.2. With regards to the business question we tested both attack success prediction and number of victims predictions. For the first classification question we tested multiple models, starting with logistic regressions, basic decision trees and a random forest model, boosting methods such as AdaBoost and XGBoost, all with GridSearch were also tested. For the Victim prediction we tested linear regression, with and without polynomial features, SVM models with different kernels and finally a basic neural network as well. The best results in this phase were achieved using a RandomForest classifier, turned with GridSearch algorithm (results in Figure 5).

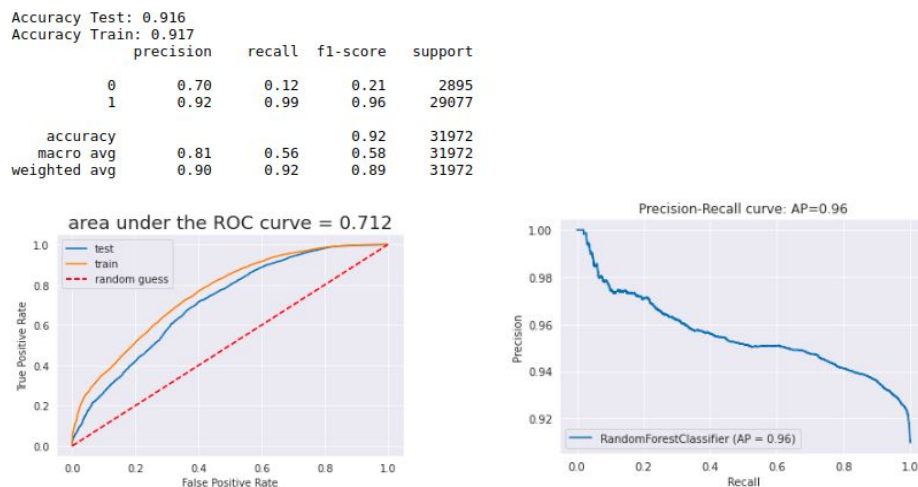


Figure 5 - RandomForest classifier with GridSearch for hyperparameter tuning

3.2.3. To further improve performance we turned to the ‘Motive’ text feature (some examples of motives from the dataset - “protest war vietnam draft”, “retaliation violence rohingya muslims myanmar”, “incident related to extortion demand”). We performed NLP preprocessing on the feature (reducing words to lemmas for example). At first glance, when testing the most common words for both successful attacks and failed attacks, not much difference was observed (‘unknown’, ‘attack’, ‘motive’, ‘specific’ and ‘however’ were the top five for both).

However, we continued the experiment with both CountVectorizer and TfidfVectorizer and tested a RandomForest Classifier and XGBoost solely on these ‘word-features’ provided by the different vectorizing methods, in order to verify that these features alone can provide a model which is better than a ‘dummy’ classifier (ROC curve of these word models in Figure 6). These models also provided ‘feature importance’ plots that can help understand which words are more informative for the model, such as ‘ethnic’, ‘sectarian’, ‘presidential’, ‘election’, ‘liberation’, ‘bomb’, ‘informer’ or ‘islam’ being some of the interesting observations.

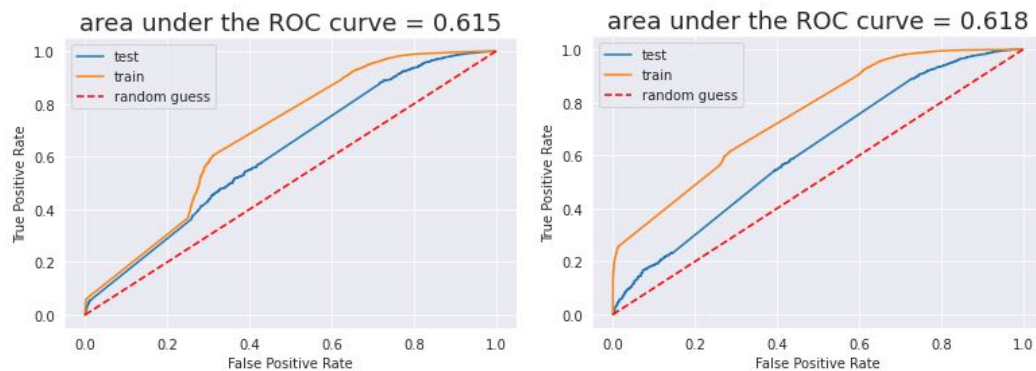




Figure 6 - RandomForest word-model (left) and XGBoost word-model (right)

Once we proved to ourselves that the ‘word-features’ add information to the model we tested the RandomForest and XGBoost models again with new GridSearches and found that this time the latter yielded the best results together with the TfidfVectorizer method.

Given the obvious correlation between some of the features and word features (for example ‘bomb’ and the explosive weapon type) we decided to apply PCA dimensionality reduction (so not all 500 vectorized words will be taken into account) the final results detailed in section 3.1 were finally achieved. The PCA represents a trade-off between interpretability of the models decision process and performance and complexity of the model.


3.2.4. Deployment - We've created an app, which can receive an input of the features and return the probability that the attack would be carried out successfully.


Will the event be extended (over 24 hours long): 


Will the event occur in the vicinity of a city ? : 

Attack Type: 

Target Type: 

Weapon Type: 

Will it be on international scale: 

Will the perpetrator(s) commit suicide: 

Perpetrator motive:

Predict

Chance of a successful attack: 92.14 %

4. Conclusions

While the final metrics of the model reflect good results and accurate predictions on a technical standpoint, it is our conclusion that the model has not done as well on the business strategy front. A model capable of providing the most accurate predictions is worthless if those predictions are not actionable and in our particular case, predicting the failure of a specific terror attack does not imply the attack should be disregarded and countermeasures need not be taken, as it's possible, that these very things are the actual causes for an attack resulting in failure and regardless, on a matter such as this, all risks should be avoided. This shortcoming in business foresight is somewhat mitigated by the 'feature importance' information provided by the model - the ability to explain in detail what makes a specific attack more or less dangerous can be valuable information, but possibly not enough to carry a business strategy.

5. Challenges

- 5.1. Dealing with an immense dataset with a very large number of features (around 135) was a certain challenge. At first glance, an expensive dataset appears to be a net positive, but we learned it requires commitment to understanding each and every feature, not only its distribution and other metrics as part of the EDA but even before that, how each feature is actually defined and how the data was collected (for example, our target, success of an attack, being defined differently for each type of terror attack).
- 5.2. As a result of the fact that a dimension-reduced model can't be properly explained, we had to choose between performance and interpretability.
- 5.3. Deploying the model was a challenge too, because we had to integrate pre-processing steps (NLP and Dimensionality Reduction) on the app level.

6. Lessons Learned

If we were to start the project over from the beginning, we might have spent more time defining the project business question and considering the final product and how it will use the model. Instead of rushing to build a model that fits that data it is important to consider the desired result and fit the model to the data and not the other way around. We learned that data isn't enough and an expensive dataset is no guarantee for a successful model. Before charging head first into data and into a project, it is important to consider the value our model can theoretically provide and not try to make predictions that will end up not being actionable (no actions or decisions can be made on their basis, accurate as they may be).