Predicting the success of cyber-related Terrorist Attacks

- Using the Global Terrorism Database provided by START at the University of Maryland:

http://www.start.umd.edu/gtd/

Citation: National Consortium For The Study Of Terrorism And Responses To Terrorism (START). (2018). Global Terrorism Database [Data File]. Retrieved From Https://Www.Start.Umd.Edu/Gtd

Research Questions and Objectives

What characterizes a successful cyber attack?

Who can benefit from any resulting information?

Government/Intelligence Agencies, Citizens, Operatives:

Safety:

- Prevent attacks; save lives.
- Identify realistic threats with precision.
- Reduce lingering consequences of attacks (infrastructure, communication, health and public welfare issues).

Understanding:

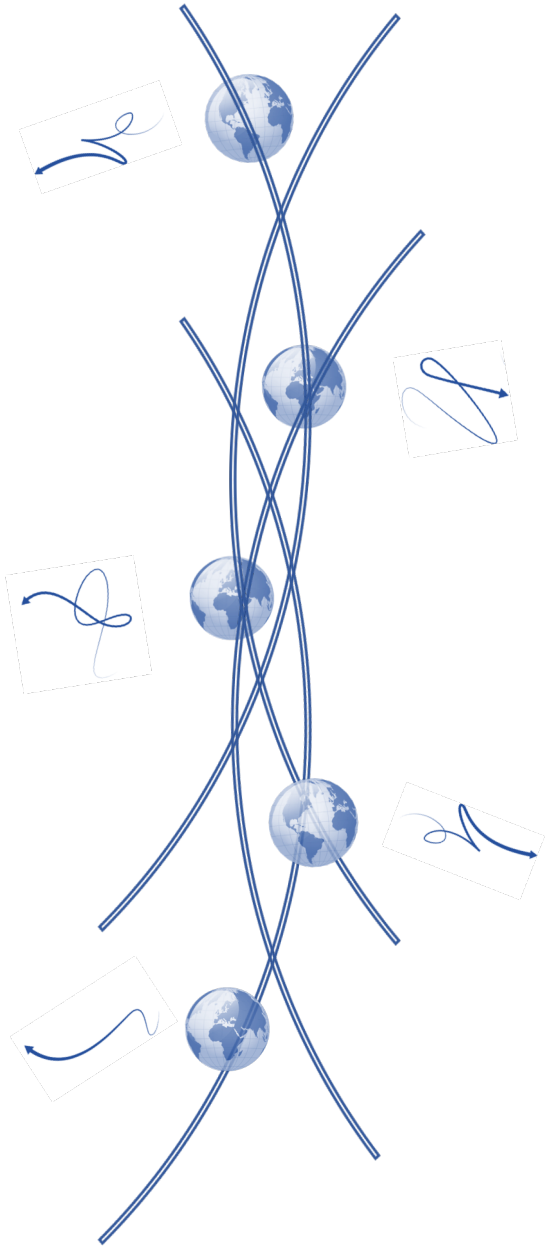
- Sharpen knowledge and accuracy regarding known and lesser-known terrorist groups, their operating territories and targets.
- Locate previously overlooked potential targets
- Increase ability to identify new/previously-unseen areas of threat.

Efficiency:

- Decrease mis-direction of labor, energy and assets
- Increase effectiveness and efficiency of resources on more realistic threats.

Background:

Hosted by the University of Maryland and funded by the US Department of State (until May 2018).



What is a terrorist attack?

Page 10 of the Codebook lists that a terrorist attack must include the following characteristics: *intention, violence and non-state-related perpetrators (i.e. those outside of the current government).*

Potential issues given the nature of this data:

Given that there are instances of terrorism which do not necessarily reach an 'enacted' state (according to this database), further researchers/people using this database (including these results) must be careful and recognize that a plethora of other instances (nearly-executed plans that went un-noticed, etc.) can and should be included for consideration when possible.

What incidents qualify for this database?:

- *"For an event to be included in the GTD, the attackers must be 'out the door,' enroute to execute the attack. Planning, reconnaissance, and acquiring supplies do not meet this threshold." p. 11 Codebook.*

Regarding our output feature 'Success':

- An attack is classified as successful if noticeable consequences occur as a result of the attack.
- *"The definition of a successful attack depends on the type of attack. Essentially, the key question is whether or not the attack type took place." p. 26 Codebook.*

Location Features:

What characterizes a successful cyber attack?

What is our output feature?

Our output feature will be the *success* column. In other words, this will be the information we are trying to predict. Given that we are building models to try and predict this information, we will use this column to compare our model predictions to some actual results and ascertain its accuracy before applying it to newer information (where we do not have information on the attack's success yet).

Other features:

This data-set consists of *numerous* features. In order to stay within the time-frame of this project, we will limit our analysis to some of the more prominent and easily recognized/comprehensible features.

This selection of features could lend itself to bias and should not be considered definitive. The use of Select K-Best and PCA algorithms below will aim to reduce some of this bias.

Feature Name	Description	Object Type	Most Frequent Value or/Mean
country_txt	Country	Object	India
region_txt	Region	Object	South Asia
city	City	Object	Unknown
target_1txt	Broad target description.	Object	Private Citizens and Property
sub_targettxt	More detailed target info.	Object	Police Security Forces/Officers
corp1	Identification of targeted corporation (if applicable).	Object	'Not Applicable'
specific_target	Specific target info (if applicable).	Object	Civilians
victim_nationalitytxt	Victim nationality.	Object	India
property	Was there property damage?	Int64	-.937

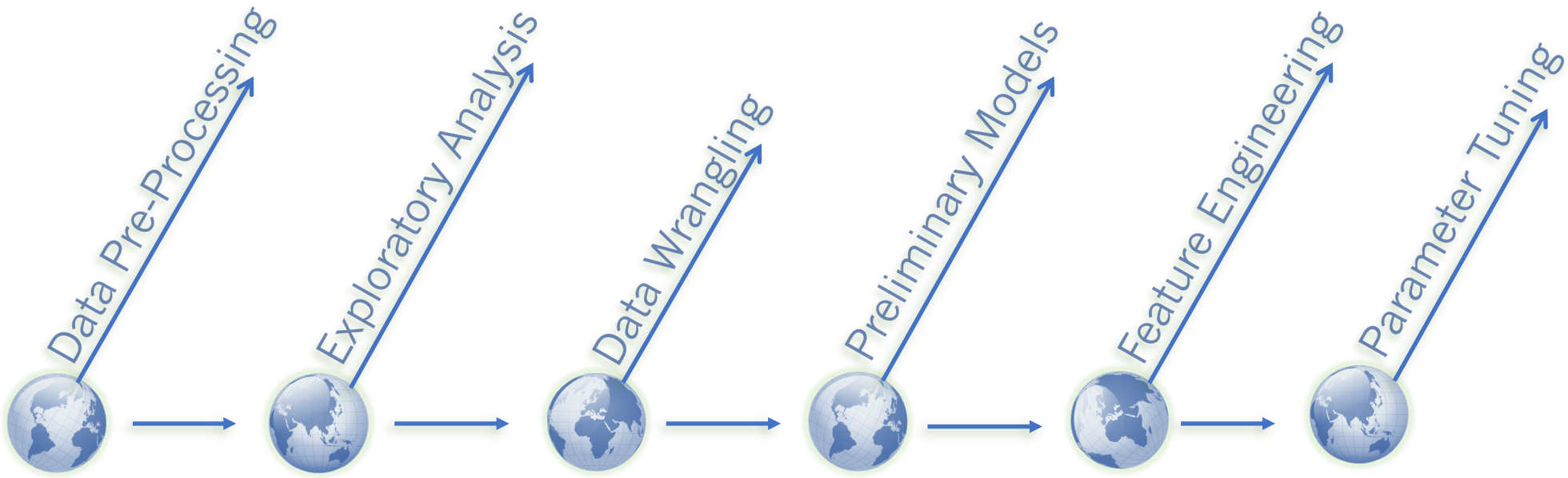
Attack Features:

Feature Names	Description	Object Type	Most Frequent Value/Mean
crit1	"Political, Economic, Religious or Social Goal?"	Int64 / *Binary*	.988
crit2	"Intention to coerce, intimidate or publicize to larger audience(s)?"	Int64 / *Binary*	.987
crit3	"Outside international humanitarian law?"	Int64 / *Binary*	.948
doubt	Is there doubt regarding classifying the attack as terrorism?	Float64 / *Binary*	-.047
suicide	Was the attack a suicide attack?	Int64 / *Binary*	.040
attack_1txt	Means of attack.	Object	Bombing/Explosion
group_name	Name of perpetrating terrorist group.	Object	Unknown
group_attrib_certainty	Is group-attribution dubious?	Float64 / *Binary*	.190
unaffil_individ	Was perpetrator identified by name and acting as an individual?	Int64 / *Binary*	.005
weapon_txt	Type of weapon.	Object	Explosives
sub_weapontxt	More detailed weapon info.	Object	Unknown Gun Type
motive	Motive of attack		n/a
summary	Synopsis of attack.	Object	n/a
success	**Was the attack successful?**	**Int64 / *Binary***	**.904**

Regarding our output feature 'Success':

- An attack is classified as successful if noticeable consequences occur as a result of the attack. (*p. 26 Codebook*).
- *"The definition of a successful attack depends on the type of attack. Essentially, the key question is whether or not the attack type took place." p. 26 Codebook.*

Workflow


Data Pre-Processing
Exploratory Analysis
Data Wrangling
Preliminary Models
Feature Engineering
Parameter Tuning

Data Pre-Processing:

Sifting out Cyber-Related Data:

- Database is vast and multi-faceted.
- Method: Regex
- Issues: Regex function might not catch all instances - decreasing accuracy

Splitting Train/Test Data:

- Prevents over-fitting
- Provides realistic and reliable scoring
- Produces safer models for implementation

Removing Null Values:

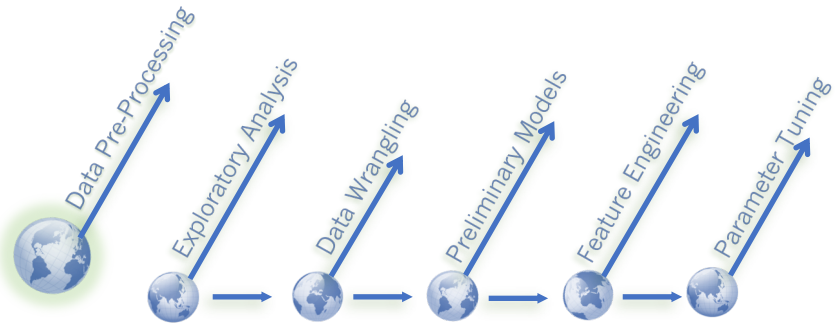
- Not a big problem after using our Regex Filter.
- Primarily focused on replacing values with 'Unknown'
- Did not have to drop too many values completely.

Selecting Primary Features:

- Focus on understandable and actionable features
- Most translatable across potential audiences
- Numerically significant (i.e. no human-assigned numeric categorizations).

Value-Filtering:

- Prepares data for later modeling
- Condenses values for feature-selection algorithm
- Keeps data comprehensible for humans while maintaining flexibility for machines/computers



Data Pre-Processing:

Sifting out Cyber-Related Data:

- Database is vast and multi-faceted (too large).
- Filter allows us to focus on cyber-instances.
- Method: Regex
- Features: *motive, summary* and *specific_target.*
- Issues: Regex function might not catch all instances - decreasing accuracy.
- Issues: Could also incorporate instances that are realistically unrelated.

Detailed Workflow:

```
# Regex Filter which fills na in this column with the value: 'Unknown'
specific = cyber_data.specific_target.str.contains(r'(internet|cell+|radio|communic+|emai+|cyb+|web|hac+)',
                                                   na = 'Unknown', flags = re.IGNORECASE)
```

```
# Combining the above results into a dataframe and looking at the shape:

cyber_data = cyber_data.loc[(cyber_data.index.isin(specific_true)) | (cyber_data.index.isin(motive_true)) |
                            (cyber_data.index.isin(summary_true)) | (cyber_data.index.isin(specific_unknown))]

cyber_data.shape
```

```
(13255, 24)
```

- Data Variable: '*cyber-data*'
- Workflow Stage: Pre-Processing
- Sub-Stage: Filtered for Cyber-Instances
- Size: 13255 Observations x 24 Features

Data Pre-Processing:

Splitting Train/Test Data:

- Prevents over-fitting.
- Provides realistic and reliable scoring.
- Produces safer models for implementation.
- Leakage Prevention: Split before further manipulation to maintain data-integrity.

  - i.e.: Any groupings/null-removals, etc. will be contained within each subset of data. This prevents data removal from leaving traces of its removal in another subset.

Detailed Workflow:

```
3  # Defining our Input and Output data:
4  # (Cleaning afterwards to prevent leakage)
5
6  X = cyber_data.drop(['event_id', 'success', 'summary', 'motive'], axis = 1)
7  Y = pd.DataFrame(cyber_data['success'])
8
9  X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = .25)
```

```
1  print(X_train.shape)
2  print(X_test.shape)
```

```
(9941, 20)
(3314, 20)
```

- Data Variable: '*X_train/X_test/Y_train/Y_test*'
- Workflow Stage: Pre-Processing
- Sub-Stage: Splitting training/test sets.
- Train-Size: 9941/3314 Observations x 20 Features

Data Pre-Processing:

Replacing Null Values:

- Not a big problem after using our Regex Filter.
- Primarily focused on replacing values in text columns with 'Unknown':
  - This maintains the rest of the row's information in the dataset without providing misleading information (such as an n/a or '0' value) for this specific column
- Only numeric column was *group_attrib_certainty.*
  - This was a very low number. Dropping these values before modeling will still maintain data integrity.

Detailed Workflow:

Select comprehensible text features:

	0
city	26
sub_targettxt	585
corp1	1671
specific_target	478
victim_nationalitytxt	103
group_attrib_crtainty	40
sub_weapontxt	1051

	0
city	6
sub_targettxt	202
corp1	578
specific_target	158
victim_nationalitytxt	24
group_attrib_crtainty	9
sub_weapontxt	344

Fill object columns with 'Unknown.'

Drop numeric nulls just before modeling.

```
1 print(cyber_train_X.shape)
2 print(cyber_test_X.shape)
```

```
(9941, 19)
(3314, 19)
```

- Data Variable: '*cyber_train/test_X, cyber_train/test_Y* '
- Workflow Stage: Pre-Processing
- Sub-Stage: Replacing nulls and selecting meaningful features.
- Size: 9941/3314 Observations x 19 Features

Selecting Primary Features:

- Focus on understandable and actionable features.
- Most translatable across potential audiences.
- Numerically significant (i.e. no human-assigned numeric categorizations).

Data Pre-Processing:

Value-Filtering:

- Prepares data for later modeling.
- Condenses values for feature-selection algorithm.
- Keeps data comprehensible for humans while maintaining flexibility for machines/computers.
- Method: Pandas .replace()
- Features: *city, group_name, specific_target, sub_target*
- Issues:
  - Not safe from human bias.
  - Highly dependent on careful research.

```python
# Here is the function we will use to take our list of values and replace it with specific group/category names:
    # We will do this for each of the test and training sets:

def magic_value_replacer(df, column, variable, string):

    df[column] = df[column].replace(variable, string)

```

sub_targettxt:

```python
internet_comm_information = ['Newspaper Journalist/Staff/Facility', 'Radio Journalist/Staff/Facility',
                             'Television Journalist/Staff/Facility', 'Other (including online news agencies)',
                             'Radio', 'Internet Infrastructure', 'Television', 'Electricity',
                             'Telephone/Telegraph']
```

specific_target:

```python
comm_related = ['Cell tower', 'Cell phone tower',  'Cell Tower', 'Cell Phone Tower', 'Cell Phone Shop',
                'Telecommunication Tower', 'Telecommunications Tower', 'Radio Stations', 'Radio Station',
                'Radio station', 'radio station', 'Radio station antenna', 'A mobile phone tower',
                'A mobile tower was targeted in the attack.', 'A Globe Telecom cell site', 'Internet Cafe',
                'Telecommunications office', 'Telecommunication Institute', 'Communications Tower',
                'Telecommunications Mast', 'An internet cafe']
polling_areas = ['Polling Station', 'Polling Center', 'Polling Stations', 'Polling Booth']
```

group_name:

```python
palestinian_separatists = ['Hamas (Islamic Resistance Movement)', 'Palestinian Islamic Jihad (PIJ)',
                           'Popular Front for the Liberation of Palestine (PFLP)', 'Popular Resistance Committees',
                           'Al-Fatah']
```

city:

```python
sunni_cities = ['Mosul', 'Kirkuk', 'Sanandaj', 'Ramadi', 'Trabzone', 'Diarbekir',
                'Damascus', 'Gwadar', 'Zahedan', 'Kandahar', 'Khiva', 'Fallujah',
                'Dakhla', 'Tajura', 'Sabrata', 'Azizia', 'Kasabat', 'Misrata', 'Tripoli',
                'Takrit', 'Tikrit']
```

Exploratory Analysis:

Class Imbalance:

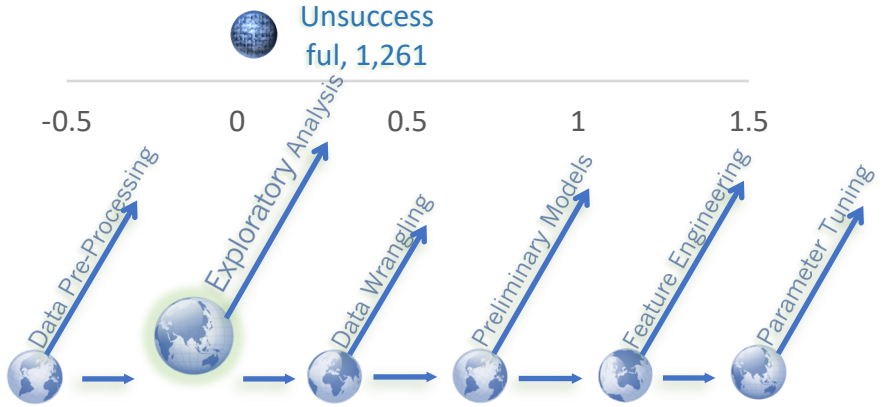
Can lead to a warped perspective:

- Lots of information about successful attacks.
- Not enough information on unsuccessful attacks.

Keeping an eye on our false positive predictions will be necessary. Most of our predictions will be positive. Limiting our incorrectly negative predictions will tell us how strong our model is.

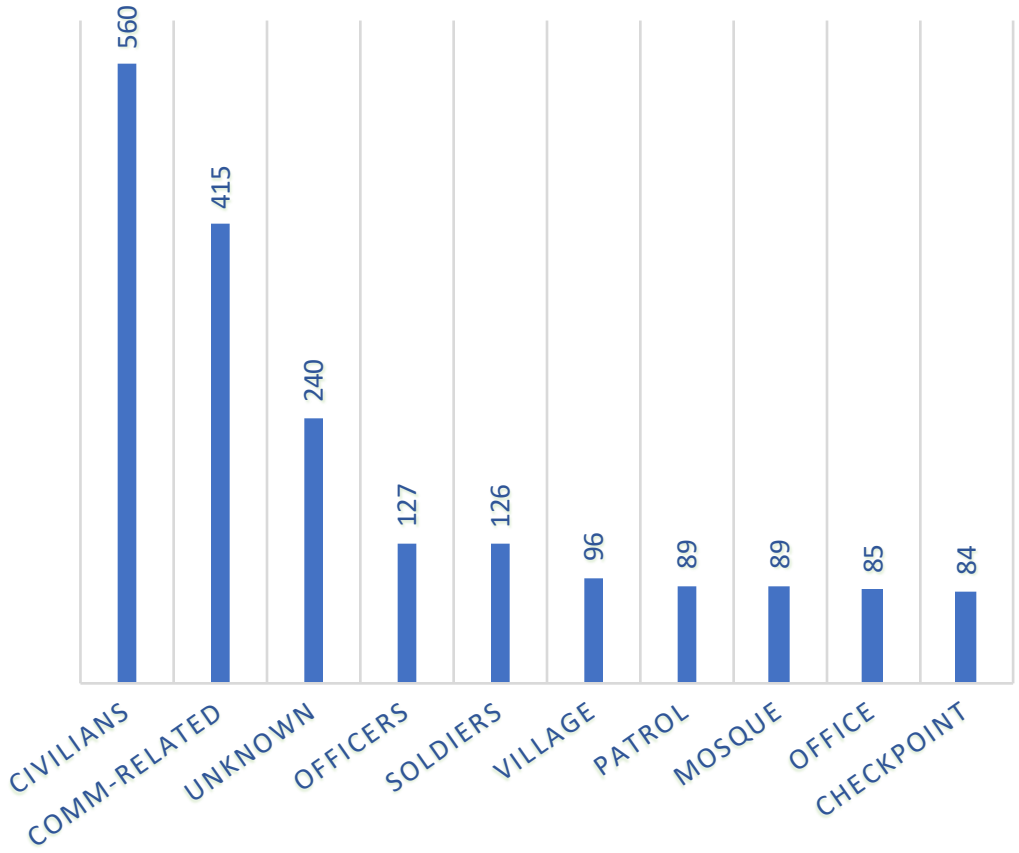
Using a balanced accuracy score will help our evaluations take class imbalance into account.

Number of Successful v. Unsuccessful Attacks

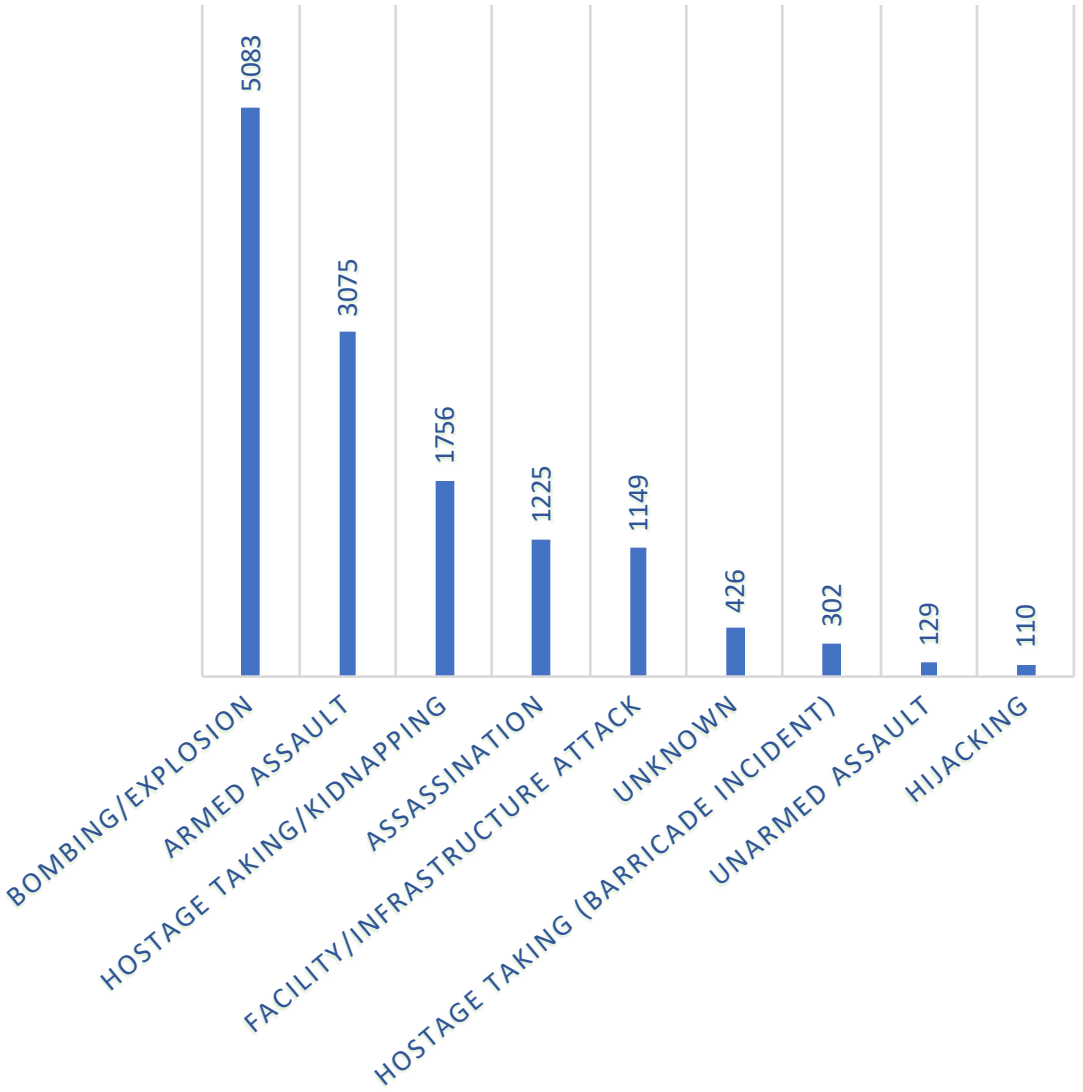


FEATURE DISTRIBUTIONS:

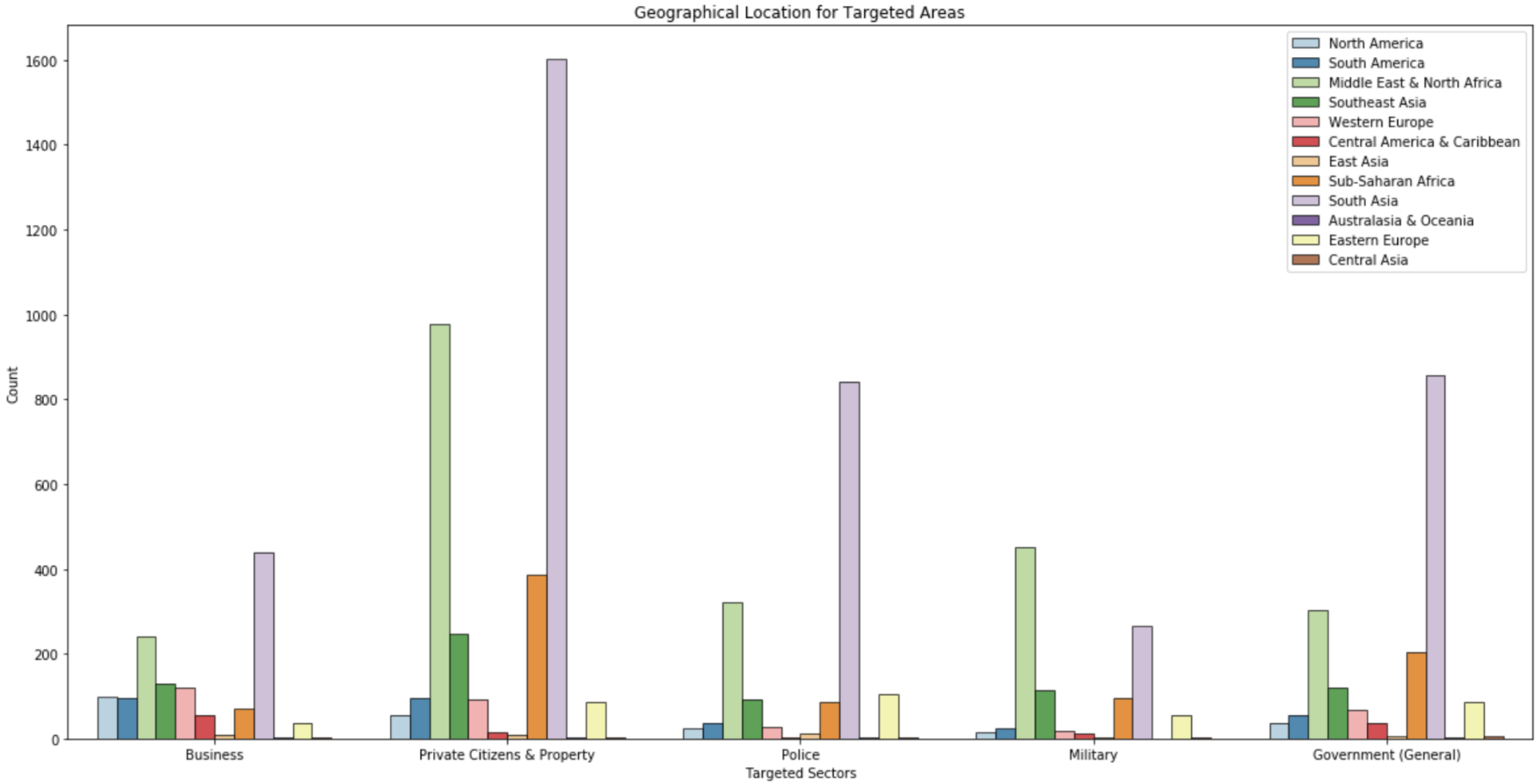
TOP 10 SPECIFIC TARGETS



TYPE OF ATTACKS

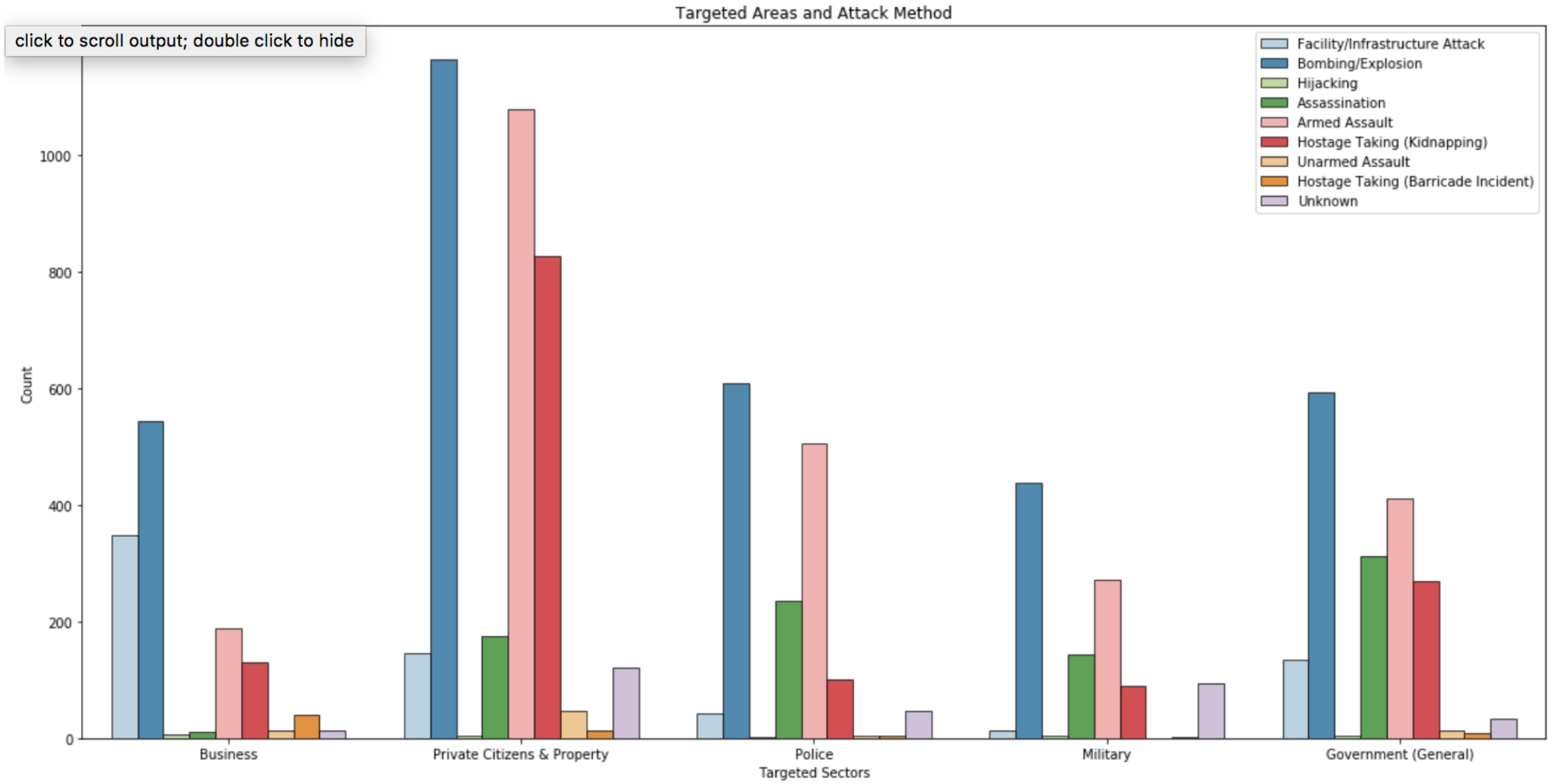


Preliminary Graphs:



This graph displays the distribution of ‘sectors’ which were targeted and how these numbers are distributed throughout the main regions of the world.

Preliminary Graphs:



This graph displays the distribution of ‘sectors’ which were targeted and how frequently these attacks were conducted with different methods.

Data Wrangling / Feature Selection

Select K-Best:

- This algorithm uses statistical measures to highlight features which have the most promising correlation to our *success* columns.
- These were the top columns selected from our dataset:

	0
0	country_txt_India
1	country_txt_Nepal
2	country_txt_Uruguay
3	country_txt_Yemen
4	attack_1txt_Armed Assault
5	attack_1txt_Assassination
6	attack_1txt_Hostage Taking (Kidnapping)
7	target_1txt_Government (General)
8	target_1txt_Private Citizens & Property
9	target_1txt_Unknown
10	sub_targettxt_Unknown

Detailed Workflow:

- Ran Select-KBest and PCA on the whole dataset
- Ran both algorithms on subsets of the data which were isolated according to the following features:

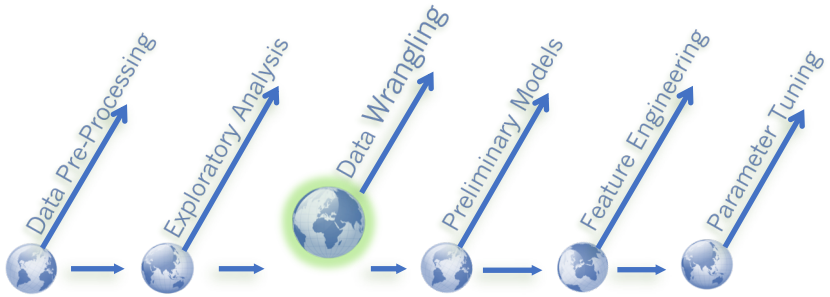
  group_name,

  city,

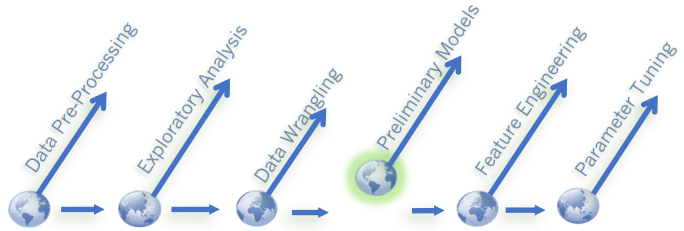
  specific_target
- Combined results from the four steps above into two specific data-frames to test with our models:
  - KBest_train/test_X/Y
  - PCA_train/test_X/Y

Principle Component Analysis:

- Takes a look at where all of the data-point lies.
  - Places vectors through areas with more variance.
  - Outputs these vectors as features to describe the data's variance.
- Difficult for humans to understand what these selected features/results mean (unlike Select K-Best).
- Can return the amount of 'explained variance,' which can be helpful.



Preliminary Models:



Here, we can see the count of the errors in each trial model prediction as well as its Balanced-Accuracy.

Why Balanced-Accuracy?

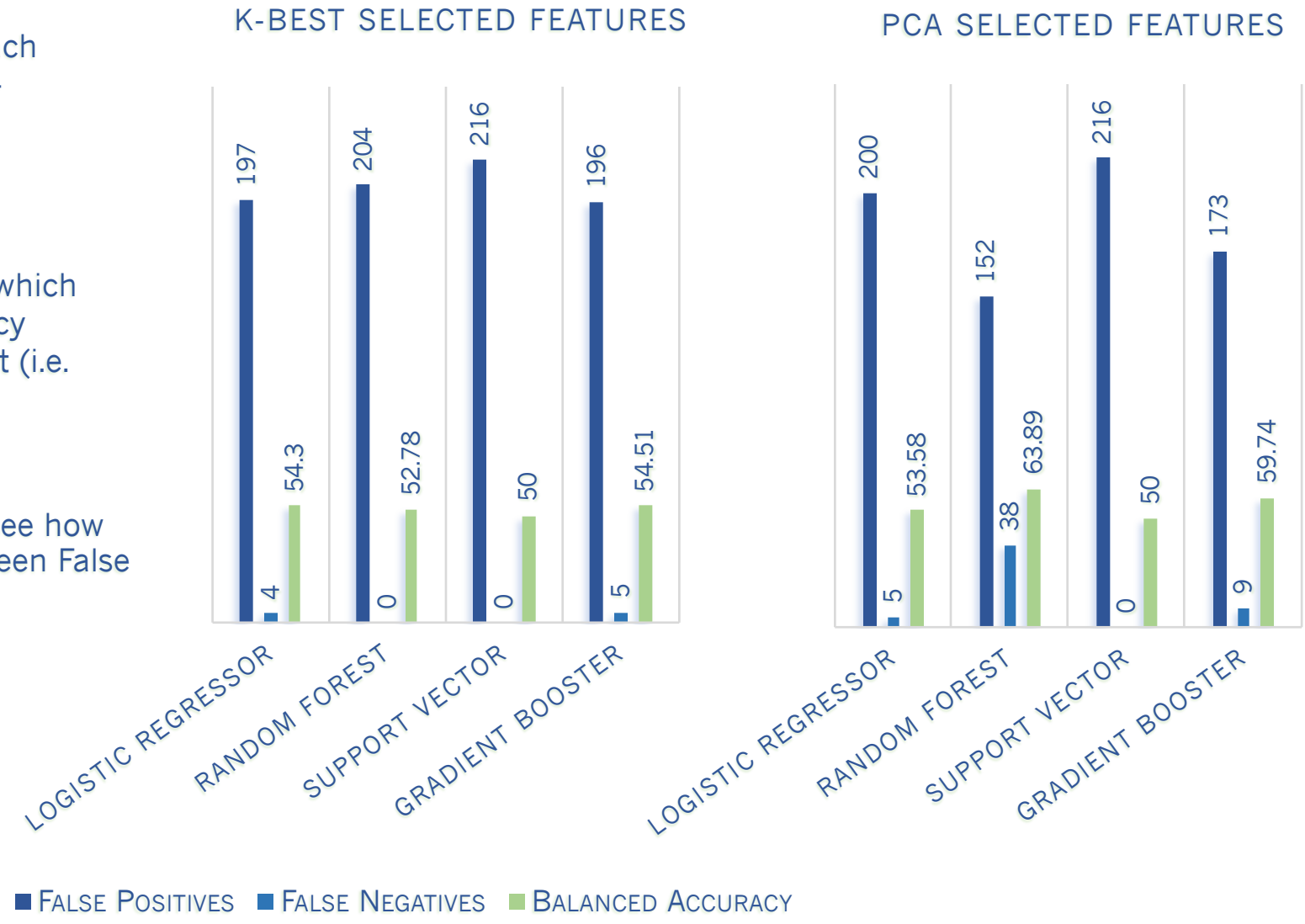
Since we have an output feature ('Success') which is highly imbalanced, we will need an accuracy score that takes class-imbalance into account (i.e. SKLearn's Balanced Accuracy metric.

This metric gives us an additional matrix to see how our two types of errors balance out (i.e. between False Positives and False Negatives).

Winner: Random Forest with K-Best.

Starting with the least False Negatives is ideal for saving lives.

Relatively low False Positives will help increase our Balanced Accuracy.



Feature Engineering:

attack_1txt: Assassination

success: 0

city: top 5:

Barometer: 0/1 (Binary)

Entire Data-set

	success
1	11994
0	1261

Data Grouped by Assassination

	success
1	606
0	310

Our previous class imbalance decreases drastically.

	success
0	27
1	16

Success rate of assassination attempts within 'Sunni_Cities.'

Our imbalance is now inverted.

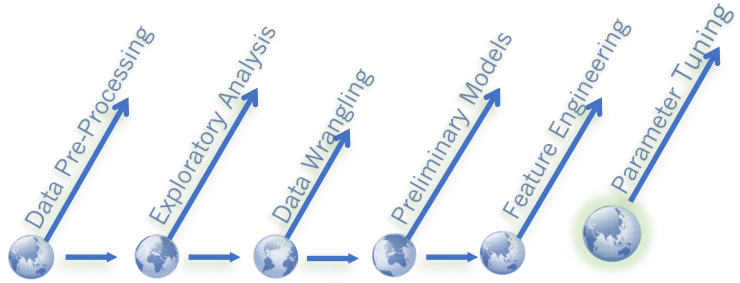
	city
Sunni_Cities	27
Unknown	16
Mogadishu	14
Split_Cities	11
Benghazi	6

Any instance in our dataset meeting these criteria (i.e. any failed assassination attempt in any of these top-5 cities), will receive a 1 in the *Barometer* column.

All else will receive a 0.

NB: Inserting other location related features into the filtering code for this engineered feature can also be done – depending on which features appear in our KBest feature set whenever we run our models.

Parameter Tuning:



Parameters tuned:

n_estimators (range: 40-280):
The number of 'trees' helping us find correct classifications.

Smaller parameters:
- Decreased accuracy.
- Decreased runtime

Larger parameters:
- Increased accuracy.
- Increased runtime
- Potential over-fitting.

*max_depth (*range: 3-15):
The 'size' of the trees in our forest.

Smaller parameters:
- Insufficient accuracy.
- Decreased runtime

Larger parameters:
- Considerable accuracy.
- Increased runtime
- Considerable over-fitting

Tuned Model Results	Max Depth	Balanced Accuracy	# of Estimators
Random Forest-Original Features (+ params)	15	56.369	80
Random Forest-Original Features (- params)	15	56.348	40
Random Forest-Engineered Features (+ params)	15	56.3	120
Random Forest-Engineered Features (- params)	15	56.46	80

NB: By looking at some more of the location features included our KBest-feature set and adding them to our filtering function for the engineered feature (see previous slide), we are able to increase our accuracy further:

- Random Forest-Original Features (+ params): 0.5837
- Random Forest-Original Features (- params: 0.5807
- Random Forest-Engineered Features (+params): 0.5933
- Random Forest-Engineered Features (- params: 0.5914

Final Summary:

Potential Weaknesses:

- More Research: The above filters are merely a start and represent a base amount of research. It could certainly serve as a starting point for someone with expertise in the field.

- Bias: the filter itself will not be able to catch all instances of cyber-related terrorism conclusively; more likely than not it will miss one certain aspect or another. Increasing the number of eyes on the model and data will help with this.

Explanatory v. Predictive Power:

- Our goal : *Prediction.*
  - We want to take *new/incoming data* and gain a prediction of whether it will be successful or not before we know the results.
  - As little variation as possible in this prediction and its accuracy score will be necessary for reliable results.

- Explanatory models: more interest to scholars or academic institutions
  - Reveals behaviors and patterns of observations that have already occurred
  - No bearing on future observations.
  - No suggestions on what these patterns might say regarding incoming data (and in this case, future terror attacks).

- Caveat: making sure that our model adapts well to new input (or test) data.
  - Allows for the variation between training and test results to be as minimal as possible (i.e. so the difference between the model's prediction and reality is minimal).
  - Predictive analysis, especially in a terror-related context, involves considerably more risk than an explanatory model, and should be handled with an intense attention to detail and accuracy.

Further Research:

Given the richness of the variables within this dataset, there are plenty of avenues for further research:

- One could use Regex to search the summary and motive columns for further detail and insight into the nature of these cyber-related tasks. The filter above is relatively broad and potentially encapsulates instances that many might not consider related to 'cyber' events.

- It would be interesting to create a more intricate regex filter which could give us a more detailed understanding of the 'cyber' aspect of these events:

    - How specifically are they related and in what manners?
    - What geographical locations tend to be hotbeds for such activity.
    - What targets are specified and why?
    - The 'motive' feature, in particular, could have extensive benefits with prediction, depending on the vocabulary used by those conducting the study and entering the data.

Final Summary:

General References:
Referenced the following sites for honing my knowledge of the models, python, etc:

https://www.analyticsvidhya.com/blog/2017/06/a-comprehensive-guide-for-linear-ridge-and-lasso-regression/
https://machinelearningmastery.com/feature-selection-machine-learning-python/
https://medium.com/@pushkarmandot/what-is-the-significance-of-c-value-in-support-vector-machine-28224e852c5a

A big shout out to Mike Swirsky for pointing out this link to me (submitted by the user 'Abdou' on Stack Overflow: https://stackoverflow.com/questions/40993626/list-memory-usage-in-ipython-and-jupyter)

For a look at the multiple resources used for filtering our data-set in the beginning stages of the work, please see the original Jupyter Notbook here.

We also used Scikit-Learn for our algorithms and modeling:
Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.