Predicting the group of Terrorists based off attack details:

Objective:

- Currently there are many terrorist attacks in the database that still have no group attributed to them if no group claimed responsibility, we wanted to create a predictive model that can possibly classify these still unknown groups?
- This project is going to be a large multiclass classifier that can use structured data and unstructured data in combination as features, to classify which terrorist group performed which.

Things to touch on:

- Data cleaning
- Accuracy on just the recent data with structured data
- Accuracy on recent data with structured and unstructured
- Accuracy on data since 1995 with structured and unstructured.
- compare random forest bow/tfidf, logistic regression with bow/tfidf compared to svm bow/tfidf

First, we have a large data set, which can be seen by visiting this link: https://www.start.umd.edu/gtd/contact/ but we also

We see the available files to us are all the excel files, and we need to decide how many we are loading in. With just the most recent, we have fairly high accuracy. With the two most recent, we have slightly lower accuracy but not too bad.
The metrics we obtained were

the Data came from the GTD otherwise known as the global terrorism databank which can be found in this link: https://www.start.umd.edu/gtd/contact/

We have: Structured data: everything but the summary data that has categorical style (i.e longitude and latitude is continous and not categorical)

Unstructured Data: the summary data which will be turned into bag of words vectors or tfidf vectors.

First we wanted to look at our unstructured data and see what types of data we had of what. Our structured data model can take in any categorical data. Our unstructured data, like a summary in text form, can be turned into bag of words or tfidf features. We can then feed that into logistic regression or support vector machines as the classifier.

- When using the two most recent (1995 -2013) and (2013-2017) We check out the two dataframes and make sure their columns are in the same in order to merge them.
- When using the two most recent (1995 -2013) and (2013-2017) We check out the two dataframes and make sure their columns are in the same in order to merge them. \

- doubtterr == 0 : we filter data frame based off of this.

- Feature Selection: We manually scrolled through , to check which data types were consistent across the board, which were all strings, which were all ints, etc. this way we will be able to feed them in easily. For feature importance

Methods for model training:

- Models trained on large dataset, data from 1995 to 2013 and then from 2013 to 2017, the same model trained on that data went down in accuracy but had a larger multi-class classification capacity.
- How model work on unstructured data like bag of words and tfidf vector features.
- In the end, we were able to combine the two features, being bag of words and the structured data

Threshold implications on our multiclass classifier list:

- When applying the threshold on the larger combined data set, we resulted in 121 labels that have sufficient training data
- When applying the threshold on the most recent dataset we resulted in 50 labels.
- Accuracy in our model went up with a higher threshold as expected

Feature selection:

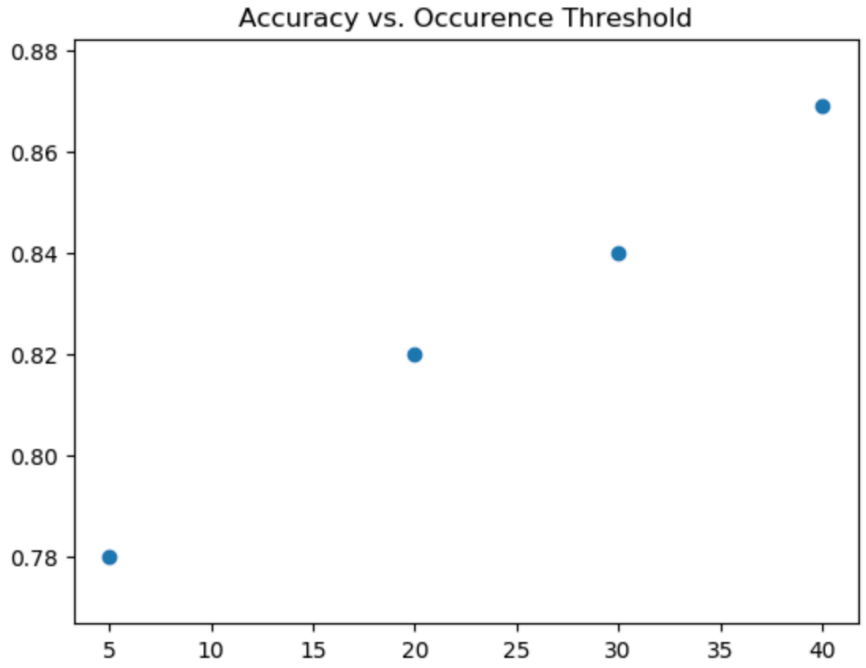
- We manually scrolled through , to check which data types were consistent across the board, which were all strings, which were all ints, etc. this way we will be able to feed them in easily. For feature importance

Methods for model training:

- Models trained on large dataset, data from 1995 to 2013 and then from 2013 to 2017, the same model trained on that data went down in accuracy but had a larger multi-class classification capacity.
- How model work on unstructured data like bag of words and tfidf vector features.
- In the end, we were able to combine the two features, being bag of words and the structured data

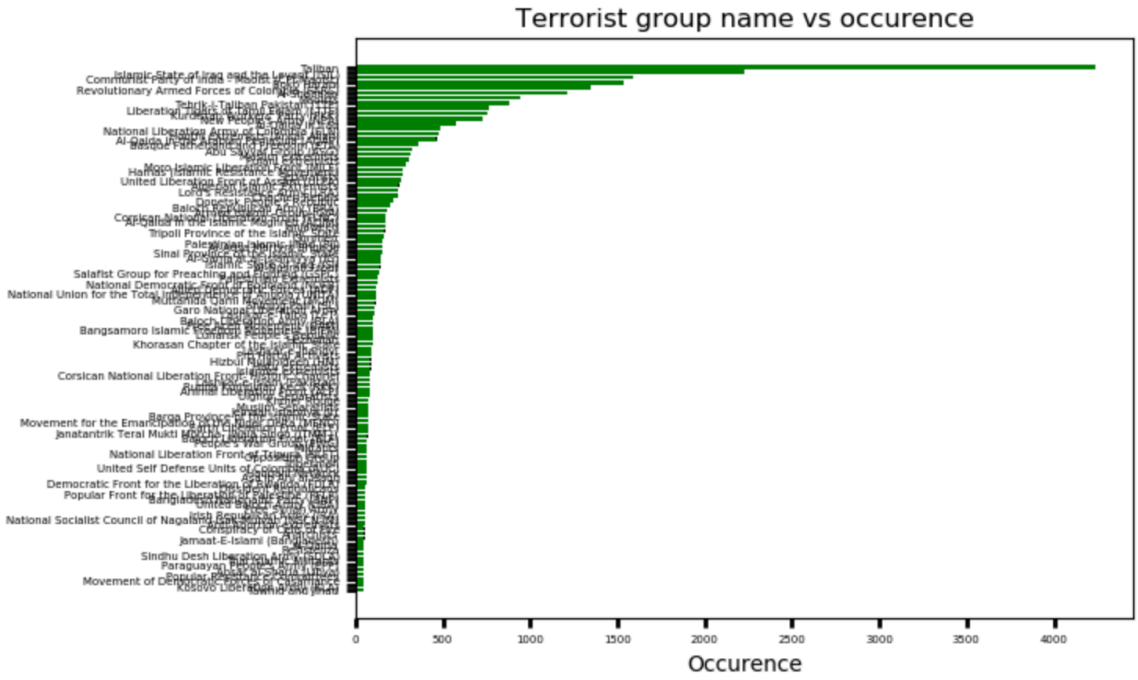
Threshold implications on our multiclass classifier list:

- When applying the threshold on the larger combined data set, we resulted in 121 labels that have sufficient training data
- When applying the threshold on the most recent dataset we resulted in 50 labels.
- Accuracy in our model went up with a higher threshold as expected

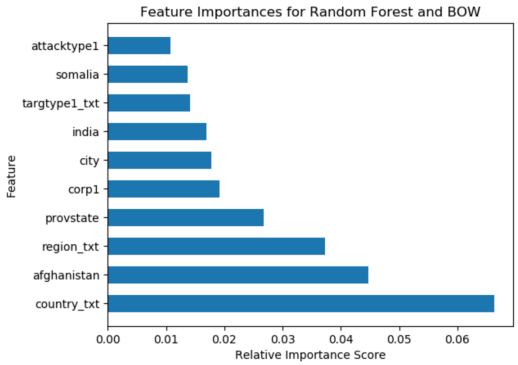


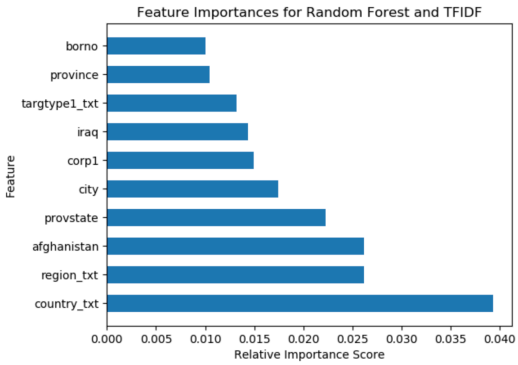
- We arbitrarily chose 30 to be the threshold number where we will discard any group names from our test-train set that only have 30 instances or more.
- We then run some tests comparing threshold numbers. As you can see, the accuracy increased linearly with the threshold.

EDA:



Top ten features in the model of random forest:





Methods for model training:

- Random Forest Classification: structured + BOW/tfidf
- Logistic Regression: structured + BOW/tfidf
- Support vector machines: structured + BOW/tfidf
- Random Forest uses label encoding for structured data
- Logistic Regression used one-hot encoding for structured data

Results for a large 121 multi-classifier data trained on data from 1994.

- We obtained Accuracy = 0.86 Random Forest Classification: structured + BOW
- We obtained Accuracy = 0.80 Random Forest Classification: structured + tfidf
- We obtained Accuracy = 0.92 for a large 121 multi-classifier data trained on data from 1994. Logistic Regression: structured + BOW
- SVM took too long to train in the end.

We found that if removed the names of the groups from the summary unstructured data, then we can run bag of words on that, because originally when we ran bag of words we were getting suspiciously high accuracy, and it was discovered that the title of group name was usually included, for example, when a group claimed responsibility, (this is also most likely the case because we filtered doubtterr = 0.)