PSG COLLEGE OF TECHNOLOGY, COIMBATORE 641004

Department of Computer Science and Engineering

19ZO02 - Social Economic Network and Analysis

Title - Terrorist Attack Network Prediction in South Asian Regions

19Z303- Aditya Sharma

19Z339 - Priyadharshini J

19Z341 - Rishika V

19Z355 - Swetha M

19Z364 - Chandhini B

PROBLEM STATEMENT

An analysis of 9 terrorist attack networks across various countries such as Afghanistan, Bangladesh, Bhutan, India, Maldives, Mauritius, Nepal, Pakistan, Sri Lanka with the goal of discovering patterns, similarities, connections, and potential relationships between one another's actors. Python was used with the Pandas and Numpy libraries to clean, manipulate, and merge all datasets. (nodes & relations). This analysis aims to find a model, using the available variables, for predicting the successfulness of terrorist attacks in South Asia. The methods used are Decision Trees and Random Forests.

DATASET DESCRIPTION

The Global Terrorist Database(GTD) is an event-level database containing more than 200,000 records of terrorist attacks that took place around the world since 1970. It is maintained by the National Consortium for the Study of Terrorism and Responses to Terrorism (START) at the University of Maryland. This codebook describes the GTD's methodology, inclusion criteria, and variables. The current GTD is the product of several phases of data collection efforts, each relying on publicly available, unclassified source materials. These include media articles and electronic news archives, and to a lesser extent, existing data sets, secondary source materials such as books and journals, and legal documents. The original set of incidents that comprise the GTD occurred between 1970 and 1997 and were collected by the Pinkerton Global Intelligence Service (PGIS)—a private security agency. After START completed digitizing these handwritten records in 2005, we collaborated with the Center for Terrorism and Intelligence Studies (CETIS) to continue data collection beyond 1997 and expand the scope of the information recorded for each attack. CETIS collected GTD data for terrorist attacks that occurred from January 1998 through March 2008, after which ongoing data collection transitioned to the Institute for the Study of Violent Groups (ISVG). ISVG continued as the primary collector of data on attacks that occurred from April 2008 through October 2011. Beginning with cases that occurred in November 2011, all ongoing GTD data collection is conducted by START staff at the University of Maryland. Users familiar with the GTD's data collection methodology are aware that incidents of terrorism from 1993 are not present in the GTD because they were lost prior to START's compilation of the GTD from multiple data collection efforts. Several efforts were made to re-collect

Table: GTD Data Collection Phases by Collection Institution

Dates of GTD Attacks	PGIS	CETIS	ISVG	START
1/1/1970 – 12/31/1997	X			X
1/1/1998 – 3/31/2008		X		X
4/1/2008 – 10/31/2011			X	X
11/1/2011 – 06/31/2020 (ongoing)				X

these incidents from original news sources. Unfortunately, due to the challenges of retrospective data collection for events that happened more than 25 years ago, the number of 1993 cases for which sources were identified is only 15% of estimated attacks. As a consequence, we exclude all 1993 attacks from the GTD data to prevent users from misinterpreting the low frequency in 1993 as an actual count.

Link to download database - Contact GTD Team (umd.edu)

TOOLS USED

- **Google colab** - To create and run our prediction model.
- **Visual Studio** - To implement the colab file locally in system.
- **Git** - Used for version control and smooth collaboration of participants.
- **Plotly Library** - Plotting interactive graphs like community graphs and sunburst plots.
- **Numpy and Pandas library**
- **Chrome browser**

CHALLENGES FACED

- Integration of Juptyer network with the huge database.
- Visualization of the huge dataset in gephi visualization.
- Understanding and coding the prediction parameters.
- Data gathering techniques and scalability issues.
- Missing data verification and correct debugging of errors.

CONTRIBUTION OF TEAM MEMBERS

Major contributions split up as follows:

Based on overall of view:

1) **Problem statement identification -** Chandhini B
2) **Data set identification -** Aditya Sharma and Priyadharshini J
3) **Database project initialization** - Rishika V
4) **Exploratory data analysis -** Aditya Sharma and Priyadharshini
5) **Documentation -** Chandhini B, Swetha M , Rishika V

Apart from this , each of us worked on the coding part . We researched on the different numpy functions and how to perform exploratory analysis. We sat and worked together in making the prediction work and the entire code.

Based on code point of view:

- **Downloading dataset and filtering dataset** - Swetha M
- **Random Forest Classification code** - Chandhini and Rishika

- **Decision Tree Code** - Aditya and Priyadharshini J

ANNEXURE I: CODE

Downloading and preprocessing of dataset.

sena.ipynb

File Edit View Insert Runtime Tools Help Last saved at 4:36 PM

Comment Share

+ Code + Text

Connect Editing

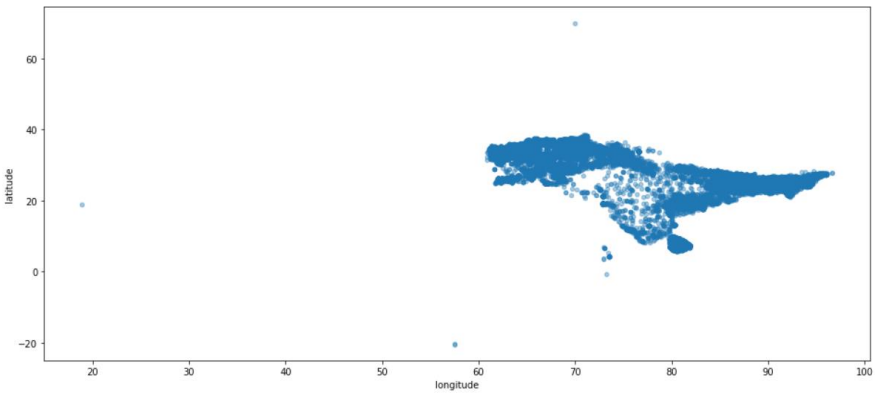
Download the dataset

```
[ ] # importing libraries
    import numpy as np
    import pandas as pd
    import plotly.graph_objs as go
    from sklearn.model_selection import train_test_split
    from sklearn import tree
    from IPython.display import Image
    from six import StringIO
    import matplotlib.pyplot as plt
    import pydotplus
    from sklearn.ensemble import RandomForestClassifier
    from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot

    from sklearn.metrics import classification_report,confusion_matrix
```

```
[ ] from google.colab import drive
    drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
[ ] #scatter plot
    df_SA.plot(kind= 'scatter', x='longitude', y='latitude', alpha=0.4, figsize=(16,7))
    plt.show()
```



Decision Tree

sena.ipynb

File Edit View Insert Runtime Tools Help Last saved at 4:39 PM

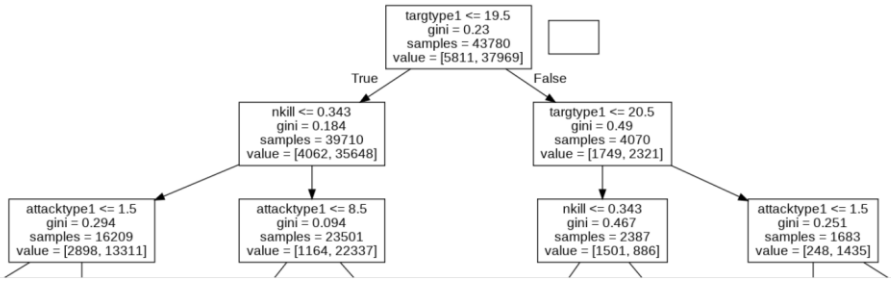
Comment Share

+ Code + Text

Connect Editing

Decision tree - max depth(3)

```
[ ] y = df_SA['success'] #this is what we're trying to predict!
    X = df_SA[features]
    dtc = tree.DecisionTreeClassifier(max_depth=3)
    dtc = dtc.fit(X_train,y_train)

    #code to print the underlying configuration
    dot_data = StringIO()
    tree.export_graphviz(dtc, out_file=dot_data,feature_names=features)
    graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
    Image(graph.create_png())
```



Random Forest Classification

Random Forest classification

+ Code + Text

```
[ ] rfclass = RandomForestClassifier(n_estimators=400)
    rfclass = rfclass.fit(X_train, y_train)
    rfclass_pred = rfclass.predict(X_test)
```

```
[ ]
    print(classification_report(y_test,rfclass_pred))
```

```
              precision    recall  f1-score   support

           0       0.82      0.52      0.63      1459
           1       0.93      0.98      0.96      9486

    accuracy                           0.92     10945
   macro avg       0.87      0.75      0.79     10945
weighted avg       0.91      0.92      0.91     10945
```

```
[ ] print(confusion_matrix(y_test,rfclass_pred))
```

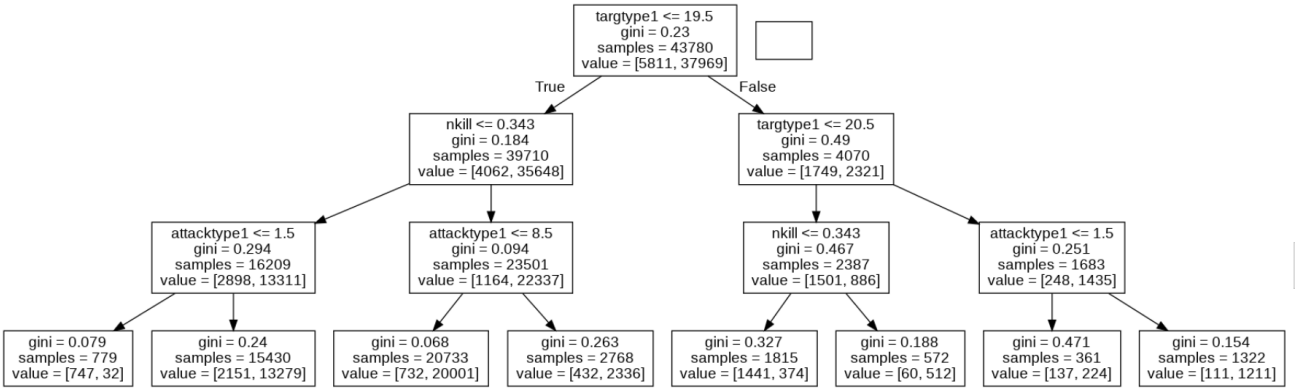
```
[[ 756  703]
 [ 169 9317]]
```

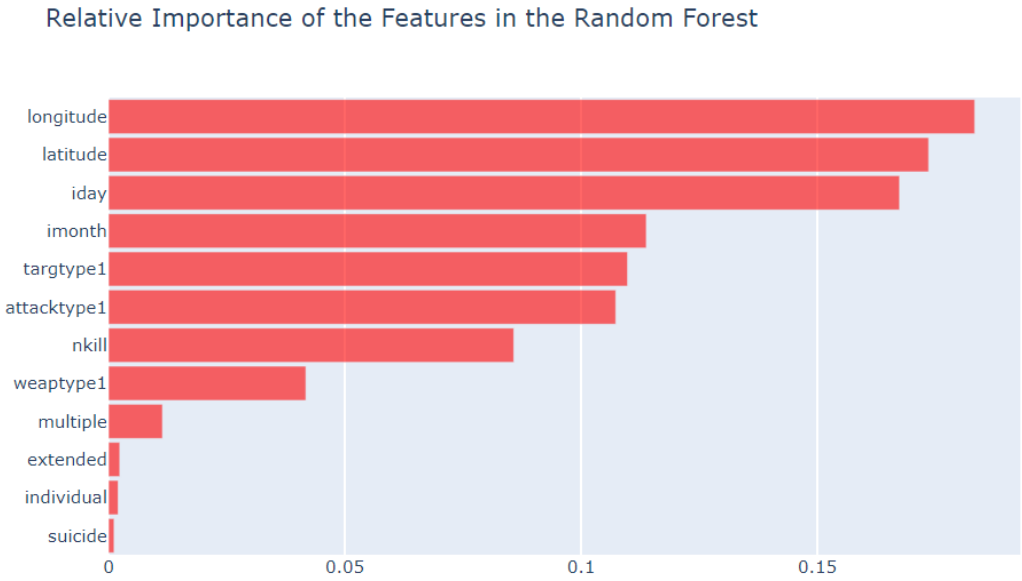
```
[ ] #listing iportance of features
    for name, score in zip(X_train[features], rfclass.feature_importances_):
        print(name, score)
```

```
succeed_or_fail = RandomForestClassifier(n_estimators=400)
succeed_or_fail = rfclass.fit(X, y) #clf

month = 12           # in which month would the attack take place
day = 23             # on which day of the month would the attack take place
extended = 0         # 1=yes, 0=no
latitude = 48.8566
longitude = 2.3522
multiple = 0         # attack is part of a multiple incident (1), or not (0)
suicide = 0          # suicide attack (1) or not (0)
attackType = 3       # 9 categories
targetType = 7       # 22 categories
individual = 0       # known group/organization (1) or not (0)
weaponType = 6       # 13 categories
nkill = 0            # number of total casualties from the attack

outcome = (succeed_or_fail.predict([[(month),(day),(extended),(latitude),(longitude),(multiple),(suicide),(attackType),(targetType),(individual),(weaponType),(nkill)]]))
if outcome == 1:
    print(outcome)
    print("The attack based on these features would be succesful.")
elif outcome == 0:
    print(outcome)
    print("The attack based on these features would NOT be succesful.")
```

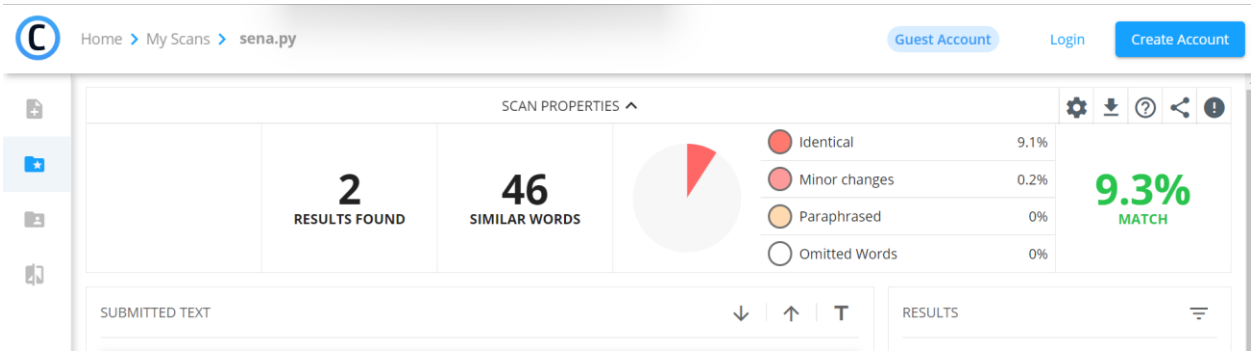
```
[1]
The attack based on these features would be succesful.
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning:

X does not have valid feature names, but RandomForestClassifier was fitted with feature names
```

ANNEXURE II: SNAPSHOTS OF THE OUTPUT





PLAGIARISM REPORT OF CODE



REFERENCES

1. Global Terrorism Database (umd.edu)
2. https://mdl.library.utoronto.ca/technology/tutorials/visualizing-network-dataset-using-gephi
3. https://libguides.brown.edu/gephi
4. https://github.com/maks-sh/Visualization-of-Global-Terrorism-Database.git
5. https://steelkiwi.com/blog/social-network-application-development-types-challenges-technologies-costs/
6. **Link to download database** - Contact GTD Team (umd.edu)