Faculty of Business

University of Moratuwa

Semester 06

DA4621– Big Data Technology Principles

Individual Assignment

Due Date of Submission

[24/08/2025]

No. of Pages

[89]

I. Acknowledgement

I want to express my heartfelt appreciation to Mr. Maninda Edirisooriya, Big Data Technology Principles Lecturer, who gave me the chance to complete this assignment. I want to express my gratitude to my friends for their assistance in completing this work within the time constraints.

II. Table of Contents

III. Table of Figures

IV. Table of Tables

01. Introduction

In recent decades, terrorism has developed as a persistent worldwide concern, threatening national security, economic stability, and public safety. Understanding the patterns, causes, and repercussions of terrorist attacks is critical for governments, politicians, scholars, and law enforcement. With the increased availability of large-scale datasets, data-driven approaches to terrorist analysis can provide valuable insights that help to improve risk assessment, early warning systems, and counter-terrorism strategies.

This assignment will investigate and analyze global terrorism patterns using the Global Terrorism Database (GTD), one of the most extensive open-source datasets on terrorist incidents in the world. The GTD provides detailed information on over 180,000 terrorist occurrences from various countries and decades, including characteristics such as attack type, target type, weapon category, fatalities, geographic location, and so on. The breadth and richness of this dataset make it ideal for big data analysis and advanced visualization techniques.

This research uses analytical methods and big data technology to uncover significant trends and hotspots in terrorist activity, evaluate temporal and spatial distributions, and investigate the human effects of terrorism around the world. The ultimate goal is to demonstrate how real-world data can be transformed into actionable knowledge that supports evidence-based decision-making in global security and policy development.

02. Problem Definition and Purpose

Problem Definition

Terrorism is a significant threat to world peace, political stability, and public safety. Despite continued international efforts, the frequency and severity of terrorist acts continue to be a worry in many places. However, given the complexity and multiple nature of terrorism, detecting patterns and high-risk locations necessitates the study of huge, multidimensional information.

This project uses the worldwide Terrorism Database (GTD) to analyze worldwide terrorism episodes from 1970 to 2017. The main problem addressed is:

"Using historical data on global terrorist incidents, analyze global terrorist incidents to identify patterns, high-risk regions, target and perpetrator profiles, and factors contributing to severe attacks, and to develop predictive models for forecasting deadly, suicide, and successful attacks, thereby enabling proactive security measures and informed counter-terrorism strategies."

So we can break the problem into several parts;

1. Identify patterns of terrorist incidents and classify types of attacks, helping to understand common modus operandi or group tactics.
2. Detect trends in attack frequency over time and forecast potential future incidents, enabling authorities to anticipate periods of increased risk.
3. Quantify the severity of attacks based on casualties and attack type, and classify incidents into low, medium, and high-risk levels for prioritization of security measures.
4. Identify high-risk geographic regions or cities where terrorist attacks are concentrated, helping allocate security resources effectively.
5. Understand which targets (e.g., government, civilians, infrastructure) are most frequently attacked and identify the most active terrorist groups, supporting targeted counter-terrorism strategies.
6. Analyze factors contributing to high-fatality events, such as attack type, weapon used, and location, to inform risk mitigation and emergency response planning.

7. Extract common narratives, keywords, and attack descriptions from incident summaries, revealing emerging threats, trends, or unusual attack tactics.
8. Predict whether an attack will be deadly (*killed > 0*).
   Predict whether an attack will be a suicide attack.
   Predict whether an attack will be successful.

This enables proactive measures and early warnings for authorities.

This analysis entails analyzing temporal trends in terrorist attacks, investigating spatial distributions across countries and regions, examining the most prevalent attack and weapon types, and determining which target classes are most commonly targeted. Furthermore, the study seeks to assess how the intensity and character of terrorism have changed over time, particularly in reaction to geopolitical events, regional wars, or counterterrorism initiatives.

Significance of the Problem

Terrorism has a severe influence on society, frequently resulting in the deaths of innocent people, destruction of infrastructure, and interruption of economic activity. In addition to the human toll, it causes political instability, displacement, and diverts state resources away from development efforts in favor of military and security solutions. Thus, assessing terrorism data is more than just an intellectual exercise; it is critical for real-world policy formation, resource allocation, and public safety planning.

The relevance of solving this challenge stems from its potential to:

- Help governments and security agencies identify and monitor high-risk places and times.
- Allow policymakers and international organizations to more efficiently allocate resources and activities.
- Help researchers and analysts create predictive models that foresee terrorism risk based on previous trends.
- Inform non-governmental organizations and humanitarian agencies working in conflict-affected areas about how to tailor aid efforts.

- Increase public and civil society understanding of the magnitude and evolving nature of terrorism around the world.

This initiative can help the battle against terrorism by translating historical incident data into actionable intelligence, allowing for more proactive, targeted, and evidence-based decision-making.

Real-World Context

Several high-profile worldwide occurrences serve to highlight the significance of this analysis. The September 11th 2001, attacks (BBC, 2001) in the United States fundamentally reshaped global security policies. The rise of extremist groups like ISIS and Boko Haram (Campbell, 2015) led to widespread violence in the Middle East and sub-Saharan Africa. Similarly, ongoing conflicts in regions such as South Asia, the Horn of Africa, and parts of Europe have resulted in recurring acts of terrorism, often with devastating humanitarian consequences (Dubale, 2024).

Using the Global Terrorism Database (GTD), a large and publicly available dataset including extensive information on over 180,000 terrorist occurrences worldwide, this study puts the problem in a real-world setting supported by empirical evidence. The GTD records several aspects of each occurrence, including the date, location, attack type, weapon used, target type, number of casualties, and responsible groups. This complexity enables a multidimensional analysis that goes beyond surface-level statistics to yield deeper insights.

Finally, this study demonstrates the importance of big data analytics in tackling socially and politically critical issues. It fills the gap between historical data and real-world action by offering a data-driven foundation for improving national and international security strategy.

03. Dataset Description

Dataset Source

This research was conducted using the Global Terrorism Database (GTD), a well-known open-source dataset managed by the National Consortium for the Study of Terrorism and Responses to Terrorism (START), which is based at the University of Maryland in the United States. The version utilized in this study was taken from Kaggle, ensuring accessibility and reproducibility for both public and academic use. The dataset can be accessed and downloaded via the following link:
Kaggle Source: https://www.kaggle.com/datasets/START-UMD/gtd/data. (Kaggle, 2018)
The GTD is one of the most extensive databases on worldwide terrorist attacks, covering both domestic and international terrorist events from 1970 to 2017, with the exception of 1993, which lacks data owing to a data loss issue.

Key Features of the Dataset

- ❖ Time Frame Covered: 1970 to 2017 (excluding 1993)
- ❖ Geographical Coverage: Worldwide (across all continents, including over 200 countries and territories)
- ❖ Unit of Analysis: Each row represents a single terrorist incident
- ❖ File Format: CSV (Comma-Separated Values)
- ❖ Dataset Size:
  - Number of Records: 181,691 incidents
  - Number of Variables (Columns): 135
  - File Size: Over 159MB (> 100MB)

Core Variables and Content Overview

The dataset includes 135structured variables per incident, covering six major categories of information. Below are the main 22 variables:

- ***Temporal Information***
  - iyear, imonth, iday: Year, month, and day of the incident
- ***Geographic Information***
  - country_txt: Country of the incident

  - region_txt: World region (e.g., South Asia, Middle East & North Africa)
  - provstate, city: Subnational administrative region and city
  - latitude, longitude: Geospatial coordinates
- ***Incident Characteristics***
  - attacktype1_txt: Method of attack (e.g., bombing, armed assault)
  - suicide: Whether the attack was a suicide bombing (1 = "Yes" The incident was a suicide attack. 0 = "No" There is no indication that the incident was a suicide)
  - success: Indicates whether the attack achieved its objective
  - summary: A brief narrative of the incident
  - motive: Claimed or suspected motive behind the attack
- ***Target Information***
  - targtype1_txt: Type of target (e.g., government, military, civilians)
  - target1: Description of the actual target entity
- ***Perpetrator Information***
  - gname: Perpetrator group (e.g., Taliban, Al-Qaeda, ISIS)
  - nperps: Number of individuals involved in the attack
  - perpetrator_kill: Number of perpetrators killed
- ***Impact and Outcome Metrics***
  - weaptype1_txt: Weapon type used (e.g., firearms, explosives)
  - nkill: Number of confirmed fatalities
  - nwound / wounded: Number of individuals wounded
  - propextent: Degree of property damage

Data Source and Compilation Methodology

The GTD is based on declassified media reports and publicly available sources. It is compiled using a rigorous, multi-stage coding procedure to assure information consistency and trustworthiness. However, due to changes in media accessibility among nations and throughout time, the writers caution against extrapolating time series trends, particularly for earlier years or under-reported locations.

The database employs a standardized definition of terrorism, classifying instances as:

“The threatened or actual use of illegal force and violence by a non-state actor to attain a political, economic, religious, or social goal through fear, coercion, or intimidation.”

Justification for Dataset Suitability

The GTD is well-suited for this undertaking for the following reasons:

1. ***Large Scale and High Dimensionality***

With almost 180,000 occurrences and 135 variables, the GTD satisfies the volume, diversity, and complexity requirements of big data. This enables complete, large-scale analytics using advanced tools and techniques including clustering, time series forecasting, and geographic mapping.

2. ***Richness and granularity of data***

The dataset contains specific incident-level information such as exact dates, locations (to the city level), types of assaults, weapons, fatalities, and accountable groups. This level of detail allows for more refined analysis, such as identifying regional hotspots, investigating group-specific behavior, and predicting casualty trends.

3. ***Global and longitudinal scope.***

The dataset spans 47 years and covers the world, allowing for both temporal and cross-regional comparisons. It enables us to identify macro patterns (e.g., global escalation or fall in terrorist activity) as well as regional shifts (e.g., increased assaults in Africa or South Asia after 2010).

4. ***Real-world Relevance***

Researchers, policy analysts, and governmental bodies (for example, the US Department of Homeland Security, the United Nations, and academic institutions) frequently mention and use the GTD. This validates the dataset's legitimacy and real-world utility, making it perfect for practical and policy-relevant study.

5. ***Applicability for Multiple Stakeholders***

Insights from this dataset can assist numerous entities, including governments and intelligence organizations (for strategic planning) ,humanitarian groups assess population threats, academic researchers (in security and conflict studies) and journalists and civic society (to raise awareness).

04. Analytical Thinking and Approach

Analytical Thinking and Approach

A structured data analysis pipeline was built to effectively assess global terrorism patterns and provide valuable insights for prevention and policy action.

Plan of Analysis

The steps below cover the entire analytical plan:

- **Phase 1: Data Preparation**
  - Imported essential Python libraries such as pyspark, pandas, numpy, matplotlib, seaborn, plotly, sklearn, nltk, and folium.
  - Uploaded the GTD dataset and inspected its structure and data types.
  - Selected key variables relevant to attack characteristics, geography, time, weapons, casualties, and group affiliations.
  - Feature Engineering to derive new features.
  - Initial EDA
- **Phase 2: Data Preprocessing**
  - Numerical Variables: Handle missing values, remove duplicates, and detect/treat outliers.
  - Categorical Variables**:** Normalize text fields (e.g., trim whitespace, standardize case).
- **Phase 3: Descriptive Analytics**
  - Summary Statistics: Generate descriptive statistics to understand distributions and central tendencies.
  - Exploratory Data Analysis (EDA)**:**
    - Charts**:** Use line charts (trends over time), bar/pie charts (attack types, target types), and choropleth maps (regional distribution), etc.
    - Interactive Visuals**:** Employ Plotly and Folium for dynamic maps and drill-down visuals.
    - I have performed exploratory data analysis (EDA) on the entire global terrorism dataset as well as a focused analysis specifically for Sri Lanka.

- **Phase 4: Advanced Data Analysis**
  - **Clustering & Incident Profiling**: Use K-Means to identify attack patterns and incident clusters.
  - **Temporal Trend Analysis & Forecasting**: Analyze attack frequency over time and build forecasting models for future predictions.
  - **Risk Scoring & Severity Index**: Compute and classify incidents into risk levels using severity scores derived from casualties and attack type.
  - **Hotspot Analysis**: Identify spatial hotspots using latitude-longitude maps and heatmaps.
  - **Target and Perpetrator Profiling**: Analyze and visualize most attacked targets and most active terrorist groups.
  - **Severity Analysis**: Understand factors contributing to high-fatality events
  - **Textual Analysis**: Use NLP techniques (tokenization, word clouds) on incident summaries to extract keywords and common attack narratives.
  - **Predictive Modeling**:
    - **Model 1**: Predict whether an attack is deadly (killed > 0)
    - **Model 2**: Predict whether an attack is a suicide attack.
    - **Model 3**: Predict whether an attack was successful.
    - Models used: Random Forest, Gradient Boosting, Logistic Regression.

Tools and Technologies

The tools and technologies used in this analysis were carefully selected to support a comprehensive and efficient data science workflow. Python was chosen as the core language due to its strong open-source ecosystem, rich libraries, and community support, making it ideal for end-to-end data analysis. **PySpark, Pandas** and **NumPy** were used for efficient data loading, manipulation, and preprocessing of structured data. For visualization, **Matplotlib**, **Seaborn**, and **Plotly** provided both static and interactive graphs, enabling in-depth exploratory data analysis. **Folium** was used to create dynamic, map-based visualizations for geographic analysis and hotspot detection. **Scikit-learn** served as the primary machine learning library for preprocessing, model training, and evaluation across various predictive tasks. To analyze unstructured text data, libraries like **NLTK** and **spaCy** were employed for natural language processing tasks such as keyword extraction and incident summary analysis. These tools collectively enabled a smooth transition across data preparation, visualization, analysis, modeling, and interpretation.

Jupyter Notebook was chosen as the development environment because of its interactive interface and ability to visualize outputs alongside code. The dataset, which was stored in CSV format, included roughly 135 columns and 181,000 records and had a size of more than 100MB. It was processed efficiently using pandas without performance concerns.

Reasoning Steps

The analysis pipeline was designed to follow a modular and logical structure, where each phase builds upon the previous to ensure clarity, depth, and actionable insights. Beginning with data preprocessing, the focus was on ensuring data quality through the treatment of missing values, duplicates, and outliers, laying a strong foundation for meaningful analysis. Exploratory Data Analysis (EDA) was then employed to identify trends, patterns, and anomalies that informed feature selection and model development. Feature engineering added analytical depth and allowed for more nuanced insights.

Clustering and profiling were employed to segment different types of incidents, facilitating a better understanding of attack patterns and geographic targeting. Temporal trend analysis and forecasting supported future risk estimation using historical data. Textual analysis of incident summaries brought qualitative dimensions into the analysis, extracting key themes from unstructured data. Finally, classification models were developed to answer critical real-world questions such as whether an attack is likely to be deadly, suicidal, or successful. This end-to-end approach ensures the analysis is both data-driven and practically useful for security agencies, policymakers, and researchers.

Assumptions, Limitations, and Constraints

Assumptions

In conducting the analysis of the Global Terrorism Database (GTD), several foundational assumptions were made to ensure a structured and meaningful interpretation of the data:

✓ **Accuracy and Representativeness of the Dataset:**

The GTD dataset is presumed to be accurate and indicative of global terrorism occurrences. The data is assumed to have been acquired using defined procedures and reliable sources such as media reports, government publications, and research groups, resulting in a realistic representation of real-world events.

✓ **Random Distribution of Missing Data:**

Missing values in several variables are presumed to occur at random and do not cause systematic bias. For example, the lack of data in perpetrator-related areas is not presumed to be concentrated in a single region or time period, which would distort the overall study.

✓ **Consistency in Attack Categorization**:

The classification of assault kinds, weapon types, target types, and perpetrator groupings is believed to be constant across areas and historical periods. This consistency is required to enable comparative analysis and the detection of patterns throughout time.

Limitations

Despite the robust nature of the GTD, several inherent limitations affect the depth and reliability of the analysis:

✓ **Potential Underreporting or Overreporting**:

The dataset relies on publicly available sources, which may vary in quality and coverage. Incidents in conflict zones or authoritarian regimes may go unreported due to censorship, a lack of media freedom, or infrastructure issues. Conversely, high-profile events may attract disproportionate attention, resulting in overrepresentation. This imbalance may mislead global comparisons and trend assessments.

- ✓ **Incomplete Temporal Coverage**:

A noteworthy shortcoming of the dataset is the entire absence of data for 1993 as a result of data loss. This gap disrupts the time series' continuity and makes it difficult to simulate long-term patterns without incorporating artifacts or assumptions about the missing year.

- ✓ **High Proportion of Missing Values in Certain Fields**:

Several potentially significant variables, including nperps (the number of perpetrators), gsubname (the perpetrators' subgroup name), and claimed (whether the attack was claimed by a group), have a large proportion of missing or unverified entries. As a result, these variables were either removed from in-depth study or handled with caution, limiting the opportunity for nuanced insights into perpetrator conduct or blame claims.

Constraints

The scope of this project was also shaped by practical, ethical, and methodological constraints:

- ✓ **Data Privacy and Ethical Considerations:**

Due to the sensitive nature of terrorism-related data, which includes information about offenders, victims, and group affiliations, the research avoided detailed profiling of specific individuals or groups. This choice was influenced by ethical data processing principles, especially as some data entries may contain mistakes or unconfirmed statements that could unfairly accuse individuals or groups.

- ✓ **Lack of Geopolitical and Socioeconomic Context**:

While the GTD provides organized event data, it can not account for the larger political, economic, or cultural contexts that drive terrorist operations. For example, spikes in violence in a certain location may correspond with civil war, foreign intervention, or socioeconomic collapse actors not included in the dataset. This limits the capacity to identify causal links or conduct policy-relevant analyses without including external data sources.z

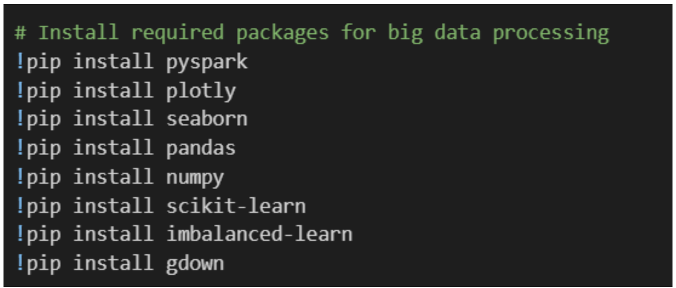
✓ **Temporal Scope Ending in 2017**:

The dataset ends in December 2017, so it does not reflect emerging trends or recent geopolitical developments such as the resurgence of the Taliban in Afghanistan, shifts in Islamic State activities, the rise of domestic terrorism in Western countries, or the role of social media in radicalization. As a result, the findings are retrospective and may not adequately reflect the present condition of worldwide terrorism.

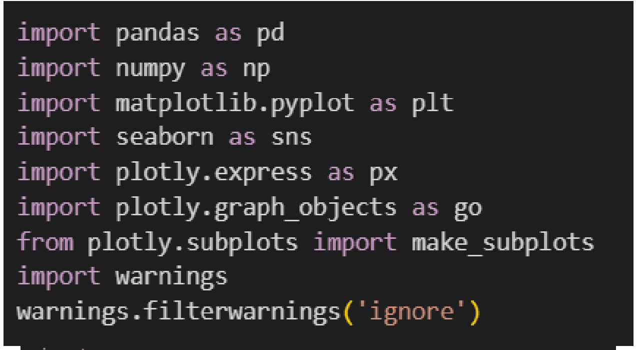
05. Phase 1: Data Preparation

To carry out this comprehensive big data analytics assignment, I imported a wide range of essential libraries that support data manipulation, preprocessing, visualization, machine learning, clustering, forecasting, and natural language processing. Core libraries like pandas, numpy, matplotlib, and seaborn were used for data handling, numerical computations, and statistical visualizations. Big data tools such as pyspark was utilized to efficiently manage and process large-scale datasets. For advanced visualizations and interactivity, libraries like plotly, folium, and missingno were employed to create insightful charts, geographic maps, and missing value patterns. Clustering and dimensionality reduction were performed using K-Means, and PCA, while time series forecasting was implemented using Facebook's Prophet. For textual analysis, libraries like CountVectorizer, wordcloud, and PIL enabled the extraction and representation of key themes from incident summaries. Machine learning models such as RandomForestClassifier, GradientBoostingClassifier, and LogisticRegression were applied to build predictive models for different classification problems. In addition, tools like StandardScaler, LabelEncoder, and train_test_split helped prepare the data for modeling, while performance metrics such as confusion_matrix and classification_report facilitated model evaluation.

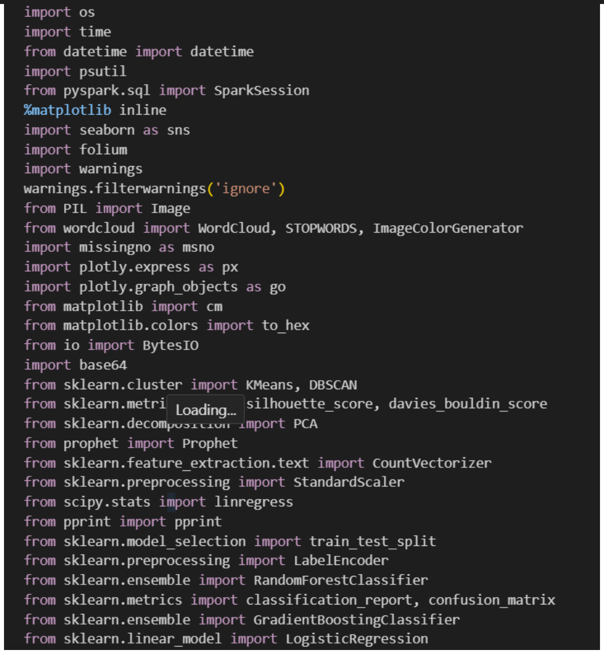
Additionally, **PySpark** was imported and a Spark session was initialized with Colab-optimized settings to handle large-scale data efficiently, enabling distributed computation, feature engineering, and model building on big datasets. This setup ensured that both exploratory and predictive analyses could be conducted effectively on the global terrorism dataset.



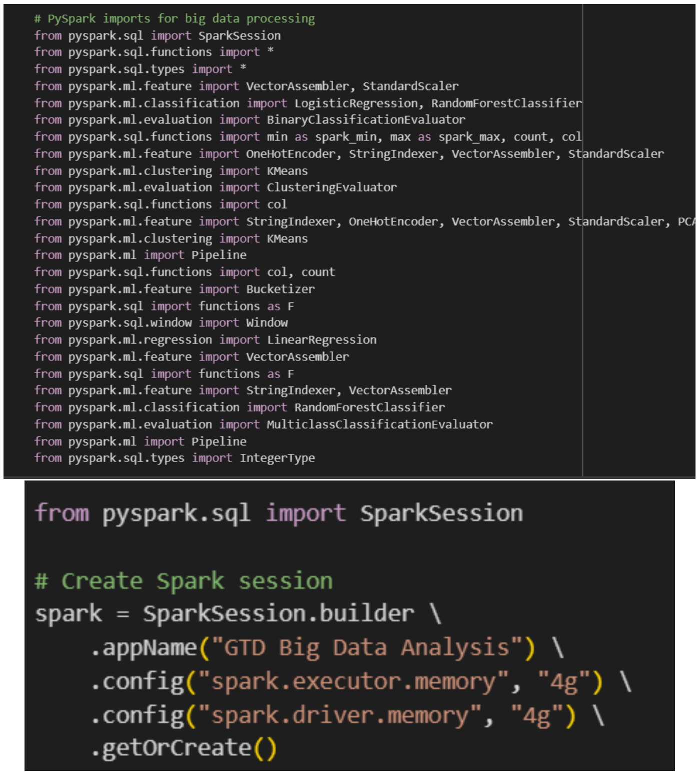
```
# Install required packages for big data processing
!pip install pyspark
!pip install plotly
!pip install seaborn
!pip install pandas
!pip install numpy
!pip install scikit-learn
!pip install imbalanced-learn
!pip install gdown
```



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import warnings
warnings.filterwarnings('ignore')
```



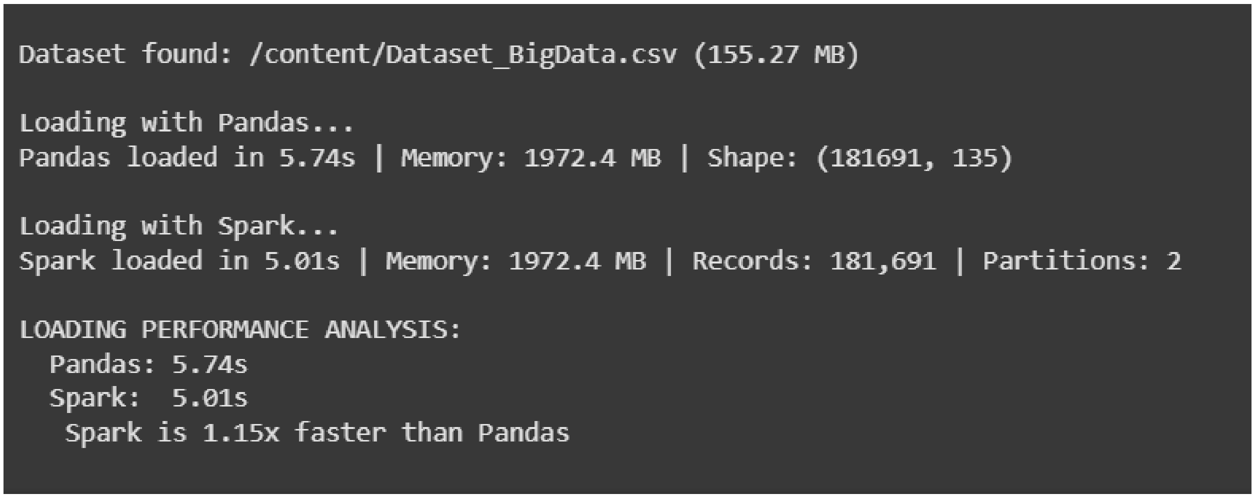
```
import os
import time
from datetime import datetime
import psutil
from pyspark.sql import SparkSession
%matplotlib inline
import seaborn as sns
import folium
import warnings
warnings.filterwarnings('ignore')
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
import missingno as msno
import plotly.express as px
import plotly.graph_objects as go
from matplotlib import cm
from matplotlib.colors import to_hex
from io import BytesIO
import base64
from sklearn.cluster import KMeans, DBSCAN
from sklearn.metri          silhouette_score, davies_bouldin_score
from sklearn.decomposition import PCA
from prophet import Prophet
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.preprocessing import StandardScaler
from scipy.stats import linregress
from pprint import pprint
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.linear_model import LogisticRegression
```



```
# PySpark imports for big data processing
from pyspark.sql import SparkSession
from pyspark.sql.functions import *
from pyspark.sql.types import *
from pyspark.ml.feature import VectorAssembler, StandardScaler
from pyspark.ml.classification import LogisticRegression, RandomForestClassifier
from pyspark.ml.evaluation import BinaryClassificationEvaluator
from pyspark.sql.functions import min as spark_min, max as spark_max, count, col
from pyspark.ml.feature import OneHotEncoder, StringIndexer, VectorAssembler, StandardScaler
from pyspark.ml.clustering import KMeans
from pyspark.ml.evaluation import ClusteringEvaluator
from pyspark.sql.functions import col
from pyspark.ml.feature import StringIndexer, OneHotEncoder, VectorAssembler, StandardScaler, PCA
from pyspark.ml.clustering import KMeans
from pyspark.ml import Pipeline
from pyspark.sql.functions import col, count
from pyspark.ml.feature import Bucketizer
from pyspark.sql import functions as F
from pyspark.sql.window import Window
from pyspark.ml.regression import LinearRegression
from pyspark.ml.feature import VectorAssembler
from pyspark.sql import functions as F
from pyspark.ml.feature import StringIndexer, VectorAssembler
from pyspark.ml.classification import RandomForestClassifier
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.ml import Pipeline
from pyspark.sql.types import IntegerType
```

```
from pyspark.sql import SparkSession

# Create Spark session
spark = SparkSession.builder \
    .appName("GTD Big Data Analysis") \
    .config("spark.executor.memory", "4g") \
    .config("spark.driver.memory", "4g") \
    .getOrCreate()
```

Figure 1 - Import Libraries

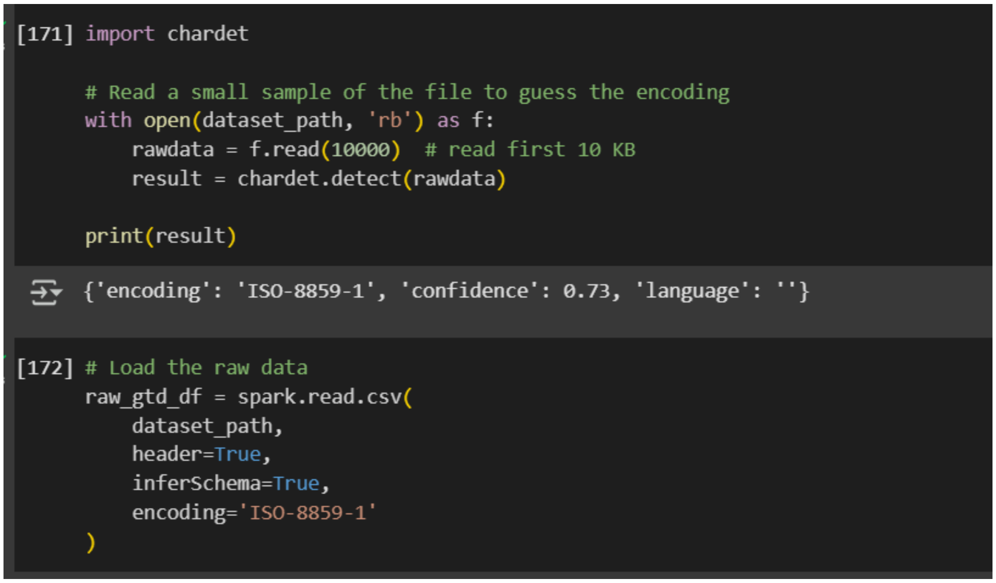


```
Dataset found: /content/Dataset_BigData.csv (155.27 MB)

Loading with Pandas...
Pandas loaded in 5.74s | Memory: 1972.4 MB | Shape: (181691, 135)

Loading with Spark...
Spark loaded in 5.01s | Memory: 1972.4 MB | Records: 181,691 | Partitions: 2

LOADING PERFORMANCE ANALYSIS:
  Pandas: 5.74s
  Spark:  5.01s
   Spark is 1.15x faster than Pandas
```

Figure 2 - Spark vs Pandas

The dataset was loaded and analyzed using both Pandas and PySpark to compare performance and resource utilization. Using Pandas, the 155 MB dataset containing 181,691 records and 135 features was loaded in 5.74 seconds with a memory usage of approximately 1,130 MB. In contrast, PySpark was able to load the same dataset in 5.01 seconds, caching the data for efficient distributed processing across 2 partitions. The performance comparison indicates that PySpark is 1.15 times faster than Pandas for this dataset, demonstrating the advantages of using Spark for handling larger datasets and performing scalable data processing in a big data environment.

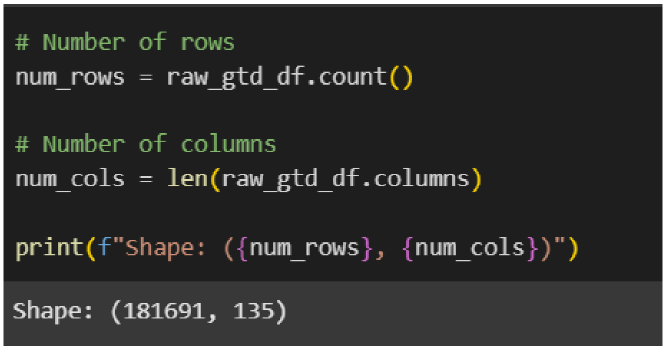
So, I will be using Spark for this assignment.



```
[171] import chardet

      # Read a small sample of the file to guess the encoding
      with open(dataset_path, 'rb') as f:
          rawdata = f.read(10000)  # read first 10 KB
          result = chardet.detect(rawdata)

      print(result)

      {'encoding': 'ISO-8859-1', 'confidence': 0.73, 'language': ''}

[172] # Load the raw data
      raw_gtd_df = spark.read.csv(
          dataset_path,
          header=True,
          inferSchema=True,
          encoding='ISO-8859-1'
      )
```

Figure 3 - File upload with Spark

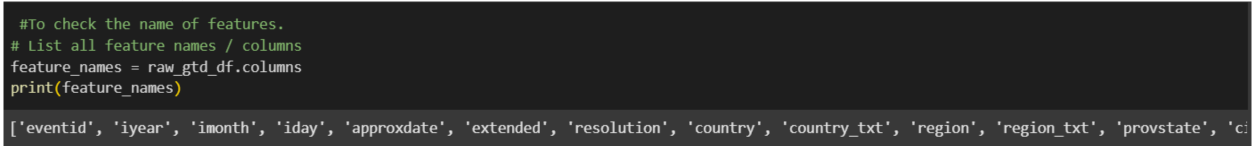
I loaded the Global Terrorism Database CSV file into a DataFrame using the specified file path and appropriate character encoding to handle special characters.



```
# Number of rows
num_rows = raw_gtd_df.count()

# Number of columns
num_cols = len(raw_gtd_df.columns)

print(f"Shape: ({num_rows}, {num_cols})")

Shape: (181691, 135)
```

Figure 4 - Data frame shape

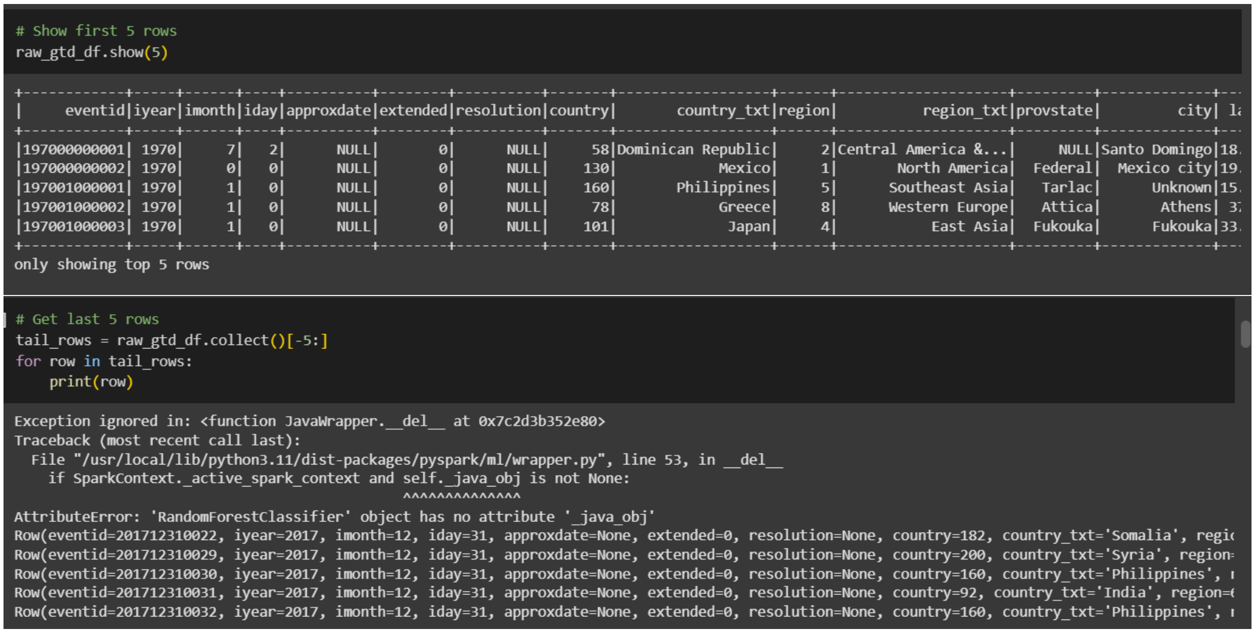
There are 181,691 rows and 135 columns. The columns are as below.



```
 #To check the name of features.
# List all feature names / columns
feature_names = raw_gtd_df.columns
print(feature_names)

['eventid', 'iyear', 'imonth', 'iday', 'approxdate', 'extended', 'resolution', 'country', 'country_txt', 'region', 'region_txt', 'provstate', 'ci
```

Figure 5 - The row/ features in the dataset

By using the head() and the tail() functions, I have gained an understanding of the data I am dealing with.



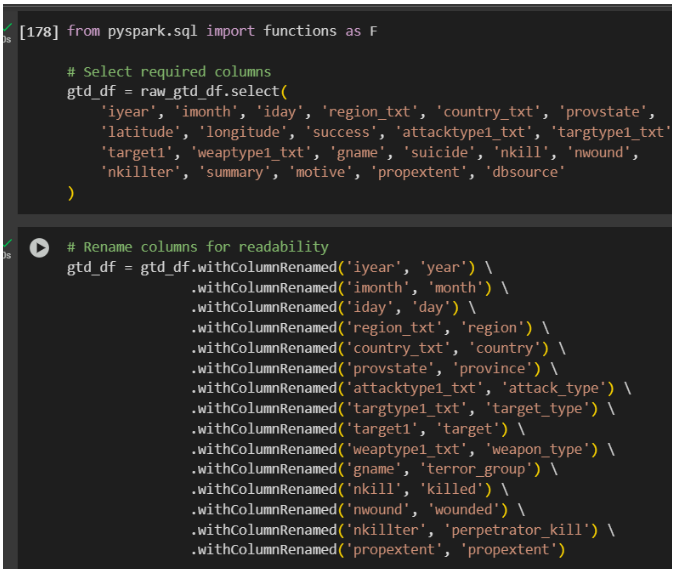
```
# Show first 5 rows
raw_gtd_df.show(5)
```

```
+------------+-----+------+----+----------+--------+----------+-------+------------------+------+--------------------+--------+-------------+---
|     eventid|iyear|imonth|iday|approxdate|extended|resolution|country|       country_txt|region|          region_txt|provstate|         city| la
+------------+-----+------+----+----------+--------+----------+-------+------------------+------+--------------------+--------+-------------+---
|197000000001| 1970|     7|   2|      NULL|       0|      NULL|     58|Dominican Republic|     2|Central America &...|    NULL|Santo Domingo|18.
|197000000002| 1970|     0|   0|      NULL|       0|      NULL|    130|            Mexico|     1|       North America| Federal|  Mexico city|19.
|197001000001| 1970|     1|   0|      NULL|       0|      NULL|    160|       Philippines|     5|      Southeast Asia|  Tarlac|      Unknown|15.
|197001000002| 1970|     1|   0|      NULL|       0|      NULL|     78|            Greece|     8|      Western Europe|  Attica|       Athens| 37
|197001000003| 1970|     1|   0|      NULL|       0|      NULL|    101|             Japan|     4|           East Asia| Fukouka|      Fukouka|33.
+------------+-----+------+----+----------+--------+----------+-------+------------------+------+--------------------+--------+-------------+---
only showing top 5 rows
```

```
# Get last 5 rows
tail_rows = raw_gtd_df.collect()[-5:]
for row in tail_rows:
    print(row)
```

```
Exception ignored in: <function JavaWrapper.__del__ at 0x7c2d3b352e80>
Traceback (most recent call last):
  File "/usr/local/lib/python3.11/dist-packages/pyspark/ml/wrapper.py", line 53, in __del__
    if SparkContext._active_spark_context and self._java_obj is not None:
                                              ^^^^^^^^^^^^^^
AttributeError: 'RandomForestClassifier' object has no attribute '_java_obj'
Row(eventid=201712310022, iyear=2017, imonth=12, iday=31, approxdate=None, extended=0, resolution=None, country=182, country_txt='Somalia', regi
Row(eventid=201712310029, iyear=2017, imonth=12, iday=31, approxdate=None, extended=0, resolution=None, country=200, country_txt='Syria', region=
Row(eventid=201712310030, iyear=2017, imonth=12, iday=31, approxdate=None, extended=0, resolution=None, country=160, country_txt='Philippines', 
Row(eventid=201712310031, iyear=2017, imonth=12, iday=31, approxdate=None, extended=0, resolution=None, country=92, country_txt='India', region=
Row(eventid=201712310032, iyear=2017, imonth=12, iday=31, approxdate=None, extended=0, resolution=None, country=160, country_txt='Philippines', 
```

Table 1 - Head and Tails of the dataset



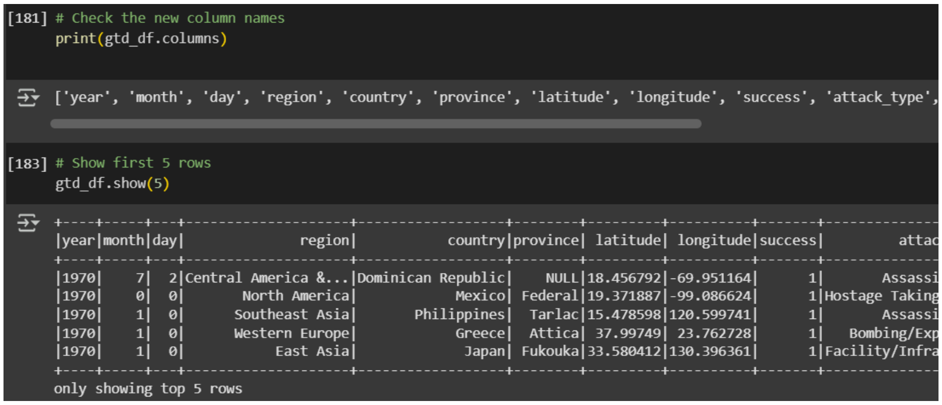
```
[178] from pyspark.sql import functions as F

      # Select required columns
      gtd_df = raw_gtd_df.select(
          'iyear', 'imonth', 'iday', 'region_txt', 'country_txt', 'provstate',
          'latitude', 'longitude', 'success', 'attacktype1_txt', 'targtype1_txt'
          'target1', 'weaptype1_txt', 'gname', 'suicide', 'nkill', 'nwound',
          'nkillter', 'summary', 'motive', 'propextent', 'dbsource'
      )
```

```
      # Rename columns for readability
      gtd_df = gtd_df.withColumnRenamed('iyear', 'year') \
                       .withColumnRenamed('imonth', 'month') \
                       .withColumnRenamed('iday', 'day') \
                       .withColumnRenamed('region_txt', 'region') \
                       .withColumnRenamed('country_txt', 'country') \
                       .withColumnRenamed('provstate', 'province') \
                       .withColumnRenamed('attacktype1_txt', 'attack_type') \
                       .withColumnRenamed('targtype1_txt', 'target_type') \
                       .withColumnRenamed('target1', 'target') \
                       .withColumnRenamed('weaptype1_txt', 'weapon_type') \
                       .withColumnRenamed('gname', 'terror_group') \
                       .withColumnRenamed('nkill', 'killed') \
                       .withColumnRenamed('nwound', 'wounded') \
                       .withColumnRenamed('nkillter', 'perpetrator_kill') \
                       .withColumnRenamed('propextent', 'propextent')
```

Figure 6 - Drop Irrelevant or Redundant Columns and getting the required columns

I then have effectively streamlined the dataset by dropping irrelevant or redundant columns and retaining only the most essential features related to the analysis of terrorist incidents. Specifically, I selected 22 key columns from the original 135, focusing on attributes such as the date and location of the attack (year, month, day, region, country, province, latitude, longitude), attack characteristics (success, attack_type, weapon_type, suicide), target and perpetrator details (target_type, target, terror_group, perpetrator_kill), casualty counts (killed, wounded), and additional contextual information (summary, motive, propextent, dbsource). I also renamed these columns to more readable and intuitive names, which improves the clarity of future analysis. As a result, the dataset was reduced to 181,691 rows and 22 columns, containing the most relevant information needed for in-depth terrorism data analysis.

The most important columns that were selected for the assignment are as below.

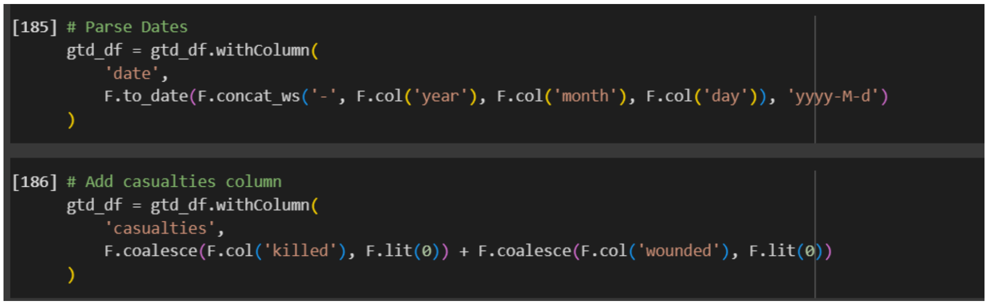


```
[181] # Check the new column names
      print(gtd_df.columns)

['year', 'month', 'day', 'region', 'country', 'province', 'latitude', 'longitude', 'success', 'attack_type',

[183] # Show first 5 rows
      gtd_df.show(5)

+----+-----+---+--------------------+------------------+--------+---------+----------+-------+------------
|year|month|day|              region|           country|province| latitude| longitude|success|        attac
+----+-----+---+--------------------+------------------+--------+---------+----------+-------+------------
|1970|    7|  2|Central America &...|Dominican Republic|    NULL|18.456792|-69.951164|      1|      Assassi
|1970|    0|  0|       North America|            Mexico| Federal|19.371887|-99.086624|      1|Hostage Taking
|1970|    1|  0|      Southeast Asia|       Philippines|  Tarlac|15.478598|120.599741|      1|      Assassi
|1970|    1|  0|      Western Europe|            Greece|  Attica| 37.99749| 23.762728|      1|   Bombing/Exp
|1970|    1|  0|           East Asia|             Japan| Fukouka|33.580412|130.396361|      1|Facility/Infra
+----+-----+---+--------------------+------------------+--------+---------+----------+-------+------------
only showing top 5 rows
```

Figure 7 - Most important features

Feature Engineering

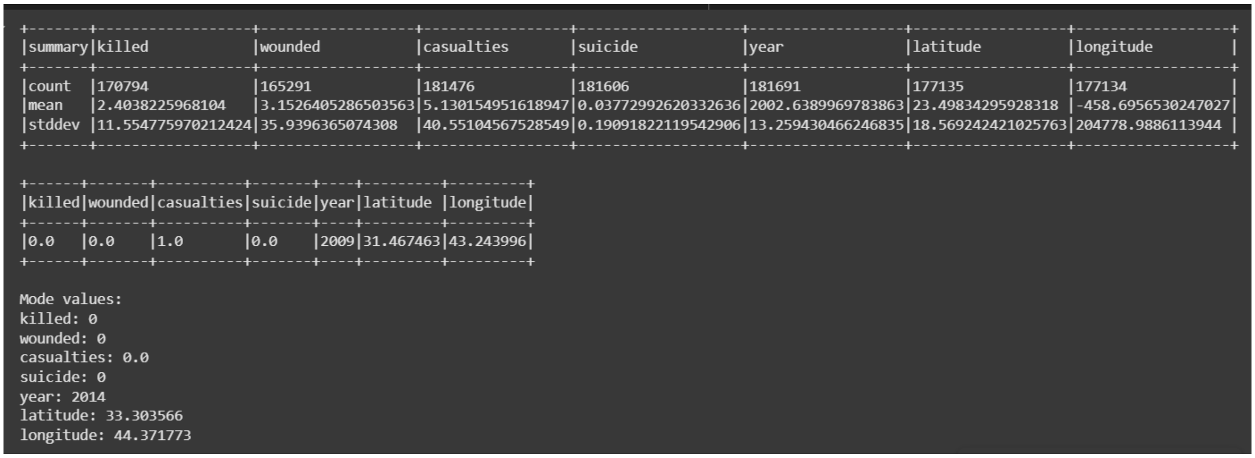
For feature engineering, I added a new column called casualties, which combines the number of people killed and wounded in each incident to provide a more comprehensive measure of the total human impact of an attack. This variable is crucial for later severity analysis, clustering, and risk scoring, as it captures the full extent of harm caused. Additionally, I parsed the date from the separate year, month, and day columns into a single date column using pd.to_datetime, enabling easier time-series analysis, trend forecasting, and chronological filtering. These feature engineering steps are important because they create new, analytically valuable variables and transform the dataset into a format that is more suitable for visualization, modeling, and insightful interpretation. This increased the no. of columns from 22 to 24.



```
[185] # Parse Dates
      gtd_df = gtd_df.withColumn(
          'date',
          F.to_date(F.concat_ws('-', F.col('year'), F.col('month'), F.col('day')), 'yyyy-M-d')
      )

[186] # Add casualties column
      gtd_df = gtd_df.withColumn(
          'casualties',
          F.coalesce(F.col('killed'), F.lit(0)) + F.coalesce(F.col('wounded'), F.lit(0))
      )
```

Figure 8 - Feature Engineering

Initial EDA



```
+-------+------------------+------------------+-----------------+-------------------+------------------+------------------+------------------+
|summary|killed            |wounded           |casualties       |suicide            |year              |latitude          |longitude         |
+-------+------------------+------------------+-----------------+-------------------+------------------+------------------+------------------+
|count  |170794            |165291            |181476           |181606             |181691            |177135            |177134            |
|mean   |2.4038225968104   |3.1526405286503563|5.130154951618947|0.03772992620332636|2002.6389969783863|23.49834295928318 |-458.6956530247027|
|stddev |11.554775970212424|35.9396365074308  |40.55104567528549|0.19091822119542906|13.259430466246835|18.569242421025763|204778.9886113944 |
+-------+------------------+------------------+-----------------+-------------------+------------------+------------------+------------------+

+------+-------+----------+-------+----+---------+---------+
|killed|wounded|casualties|suicide|year|latitude |longitude|
+------+-------+----------+-------+----+---------+---------+
|0.0   |0.0    |1.0       |0.0    |2009|31.467463|43.243996|
+------+-------+----------+-------+----+---------+---------+

Mode values:
killed: 0
wounded: 0
casualties: 0.0
suicide: 0
year: 2014
latitude: 33.303566
longitude: 44.371773
```

Figure 9 - Initial EDA

An initial exploratory analysis was conducted on the numerical variables of the Global Terrorism Database, including killed, wounded, casualties, suicide, year, latitude, and longitude. The dataset contains varying numbers of observations per variable, with counts ranging from 165,291 for wounded to 181,606 for suicide. On average, incidents resulted in approximately 2.4 deaths, 3.15 wounded, and 5.13 total casualties, with a very small proportion (0.038) involving suicide attacks. The standard deviations indicate high variability in casualties and geographic coordinates, reflecting the sporadic and widespread nature of terrorist incidents. Median values show that more than half of the incidents involved zero deaths, zero wounded, and zero suicide attacks, highlighting that many events were minor in scale. Mode analysis confirms that the most frequently reported values were zero for casualties, deaths, and wounded, with the most common year being 2014, and the most common locations concentrated around latitude 33.3 and longitude 44.37. Overall, these statistics provide a foundational understanding of the distribution, central tendency, and variability in the dataset, guiding further analysis.

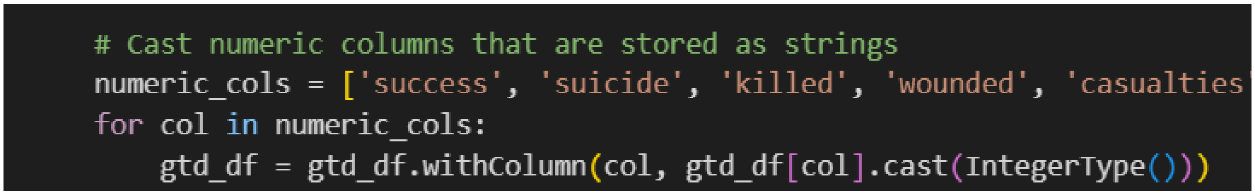
06. Phase 2: Data Preprocessing

Preprocessing for Numerical Variables

Now, for data preprocessing, I have first done the preprocessing for numerical variables.

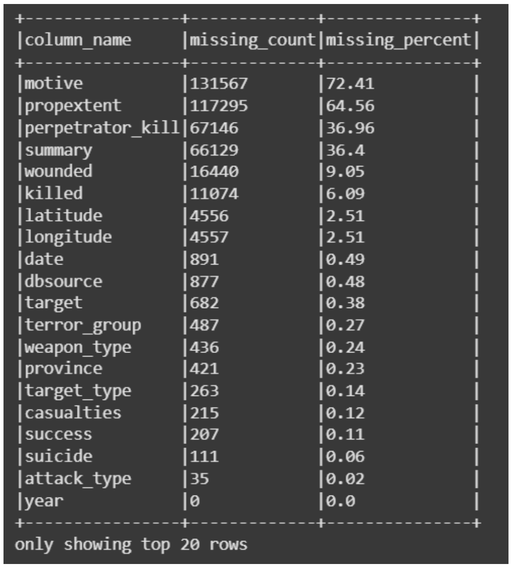
i. Handling Missing Values:

To ensure the completeness and reliability of the dataset, missing values were systematically identified and addressed. A custom function was created to quantify both the absolute and relative extent of missing values across columns.



```
# Cast numeric columns that are stored as strings
numeric_cols = ['success', 'suicide', 'killed', 'wounded', 'casualties'
for col in numeric_cols:
    gtd_df = gtd_df.withColumn(col, gtd_df[col].cast(IntegerType()))
```

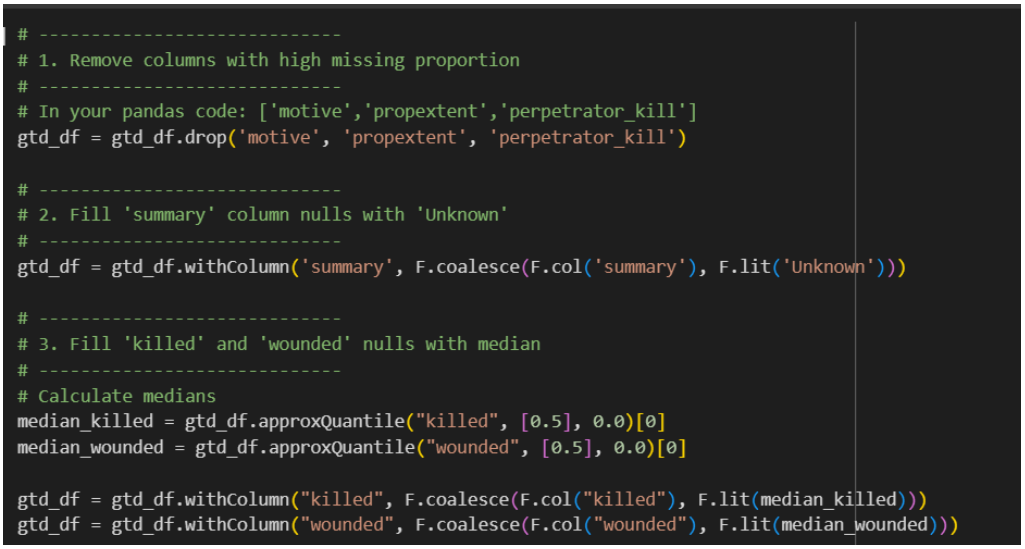
Figure 10 - Checking for missing values



```
+----------------+-------------+---------------+
|column_name     |missing_count|missing_percent|
+----------------+-------------+---------------+
|motive          |131567       |72.41          |
|propextent      |117295       |64.56          |
|perpetrator_kill|67146        |36.96          |
|summary         |66129        |36.4           |
|wounded         |16440        |9.05           |
|killed          |11074        |6.09           |
|latitude        |4556         |2.51           |
|longitude       |4557         |2.51           |
|date            |891          |0.49           |
|dbsource        |877          |0.48           |
|target          |682          |0.38           |
|terror_group    |487          |0.27           |
|weapon_type     |436          |0.24           |
|province        |421          |0.23           |
|target_type     |263          |0.14           |
|casualties      |215          |0.12           |
|success         |207          |0.11           |
|suicide         |111          |0.06           |
|attack_type     |35           |0.02           |
|year            |0            |0.0            |
+----------------+-------------+---------------+
only showing top 20 rows
```

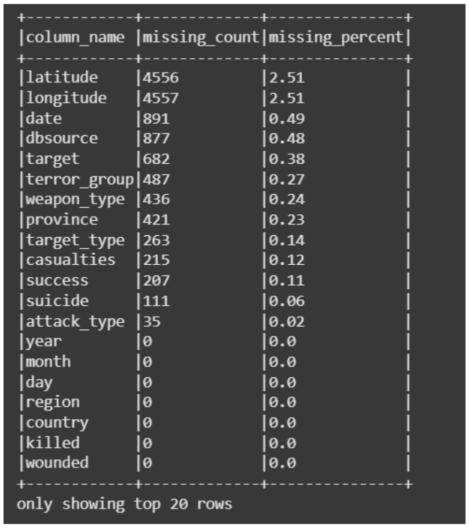
Table 2 - % of Missing Values

Based on this analysis, columns such as motive, propextent, and perpetrator_kill were removed due to having a high proportion of missing data, rendering them unsuitable for analysis. The summary column, primarily textual in nature, had numerous missing entries, which were replaced with the placeholder "Unknown" to retain the structure without introducing bias. For numerical variables like killed and wounded, where missing values were relatively sparse, median imputation was employed to maintain the distribution of the data and prevent skewness that could arise from extreme values.



```
# ----------------------------
# 1. Remove columns with high missing proportion
# ----------------------------
# In your pandas code: ['motive','propextent','perpetrator_kill']
gtd_df = gtd_df.drop('motive', 'propextent', 'perpetrator_kill')

# ----------------------------
# 2. Fill 'summary' column nulls with 'Unknown'
# ----------------------------
gtd_df = gtd_df.withColumn('summary', F.coalesce(F.col('summary'), F.lit('Unknown')))

# ----------------------------
# 3. Fill 'killed' and 'wounded' nulls with median
# ----------------------------
# Calculate medians
median_killed = gtd_df.approxQuantile("killed", [0.5], 0.0)[0]
median_wounded = gtd_df.approxQuantile("wounded", [0.5], 0.0)[0]

gtd_df = gtd_df.withColumn("killed", F.coalesce(F.col("killed"), F.lit(median_killed)))
gtd_df = gtd_df.withColumn("wounded", F.coalesce(F.col("wounded"), F.lit(median_wounded)))
```

Figure 11 - Handling Missing Values



```
+------------+-------------+---------------+
|column_name |missing_count|missing_percent|
+------------+-------------+---------------+
|latitude    |4556         |2.51           |
|longitude   |4557         |2.51           |
|date        |891          |0.49           |
|dbsource    |877          |0.48           |
|target      |682          |0.38           |
|terror_group|487          |0.27           |
|weapon_type |436          |0.24           |
|province    |421          |0.23           |
|target_type |263          |0.14           |
|casualties  |215          |0.12           |
|success     |207          |0.11           |
|suicide     |111          |0.06           |
|attack_type |35           |0.02           |
|year        |0            |0.0            |
|month       |0            |0.0            |
|day         |0            |0.0            |
|region      |0            |0.0            |
|country     |0            |0.0            |
|killed      |0            |0.0            |
|wounded     |0            |0.0            |
+------------+-------------+---------------+
only showing top 20 rows
```

Figure 12 - Missing value % after correcting for Missing Values

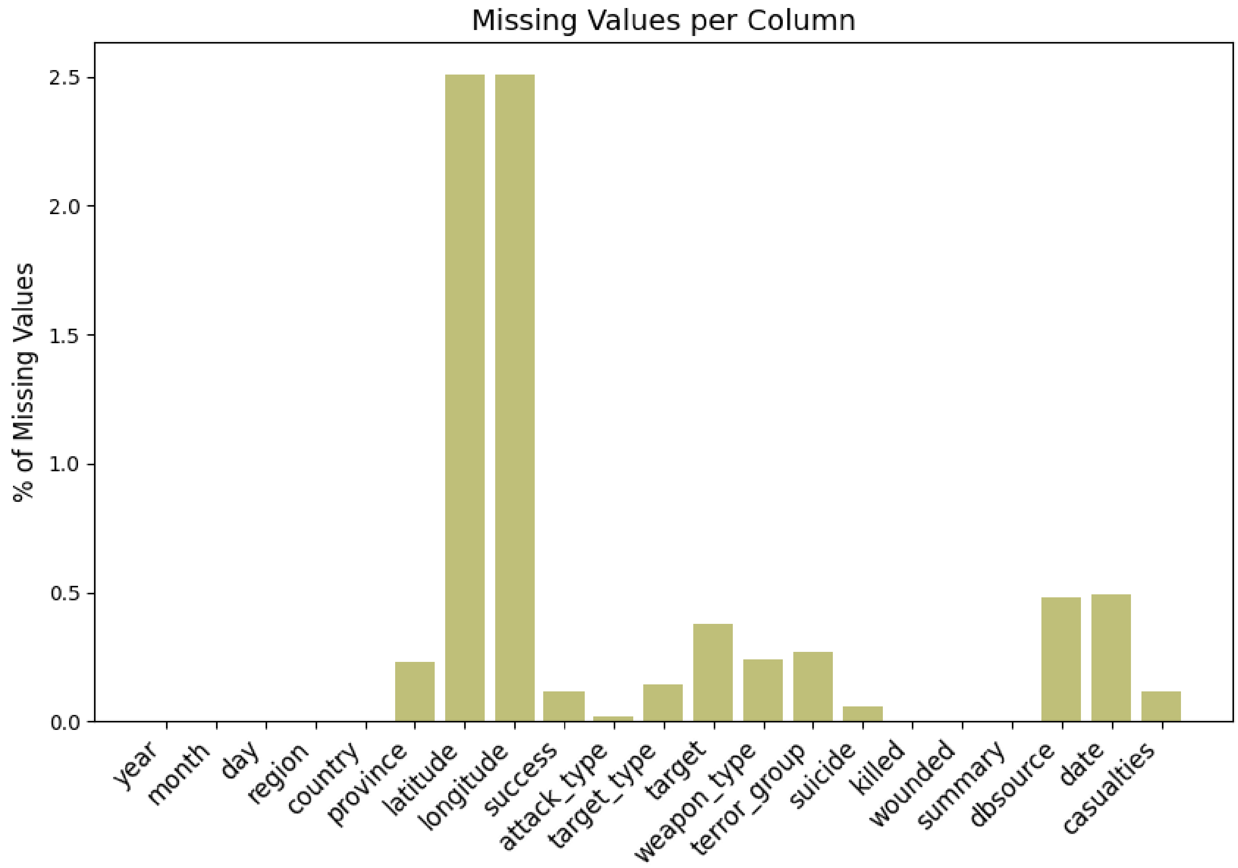
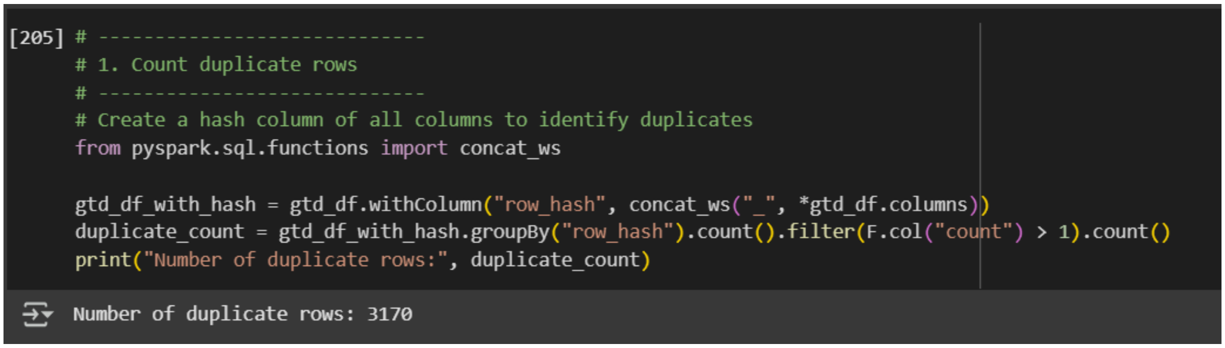


Figure 13 - Visualising the missing values

ii. *Handling Duplicates Values*

Duplicate records in the dataset can lead to biased analysis, especially in frequency-based or aggregate computations. A check for duplicated rows revealed 3,170 duplicates within the dataset.

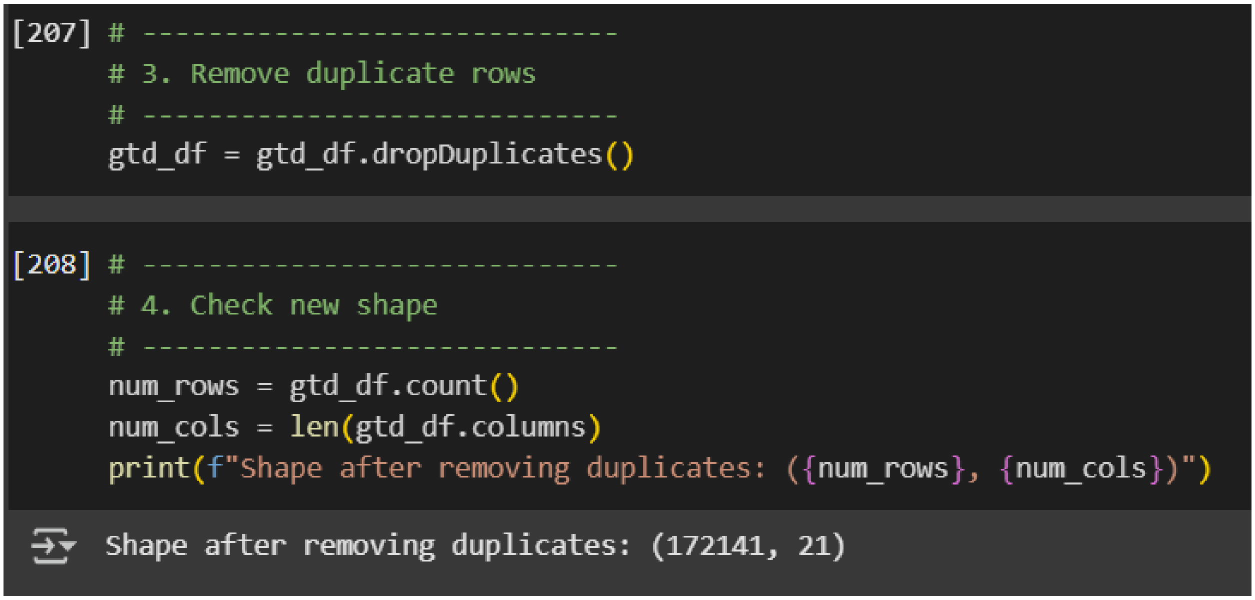


```
[205] # -----------------------------
      # 1. Count duplicate rows
      # -----------------------------
      # Create a hash column of all columns to identify duplicates
      from pyspark.sql.functions import concat_ws

      gtd_df_with_hash = gtd_df.withColumn("row_hash", concat_ws("_", *gtd_df.columns))
      duplicate_count = gtd_df_with_hash.groupBy("row_hash").count().filter(F.col("count") > 1).count()
      print("Number of duplicate rows:", duplicate_count)
```

```
Number of duplicate rows: 3170
```

Figure 14 - Check for duplicate rows

These redundant entries were promptly removed using the drop_duplicates() function. Following this operation, the number of records was reduced from 181,691 to 172,141, ensuring each incident was uniquely represented and analytical results were not distorted by repeated cases.



```
[207] # ----------------------------
      # 3. Remove duplicate rows
      # ----------------------------
      gtd_df = gtd_df.dropDuplicates()
```

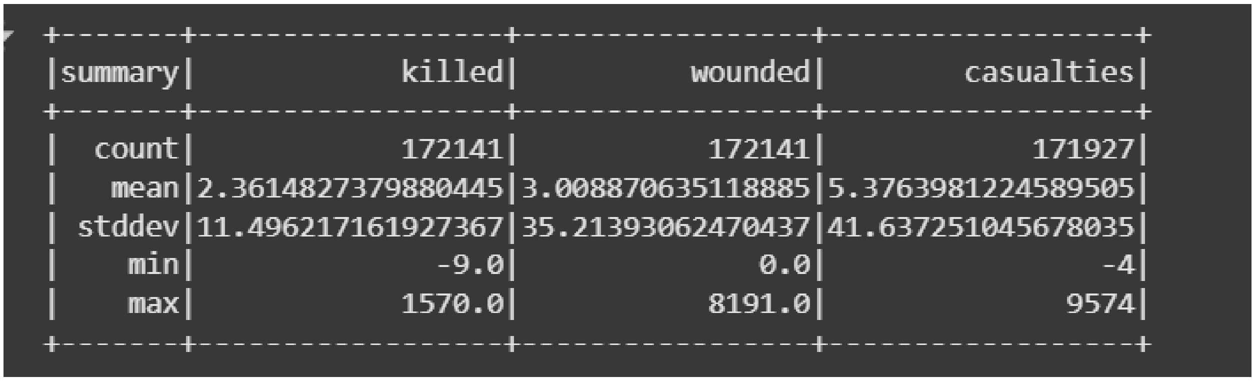
```
[208] # ----------------------------
      # 4. Check new shape
      # ----------------------------
      num_rows = gtd_df.count()
      num_cols = len(gtd_df.columns)
      print(f"Shape after removing duplicates: ({num_rows}, {num_cols})")
```

```
Shape after removing duplicates: (172141, 21)
```

Figure 15 - Removing Duplicate Rows

iii. Handling Outliers

Outliers in numerical features like killed, wounded, and casualties can disproportionately influence statistical analyses and model performance.



```
+-------+------------------+-----------------+------------------+
|summary|            killed|          wounded|        casualties|
+-------+------------------+-----------------+------------------+
|  count|            172141|           172141|            171927|
|   mean|2.3614827379880445|3.008870635118885|5.3763981224589505|
| stddev|11.496217161927367|35.21393062470437|41.637251045678035|
|    min|              -9.0|              0.0|                -4|
|    max|            1570.0|           8191.0|              9574|
+-------+------------------+-----------------+------------------+
```

Table 3 - Outlier Detection

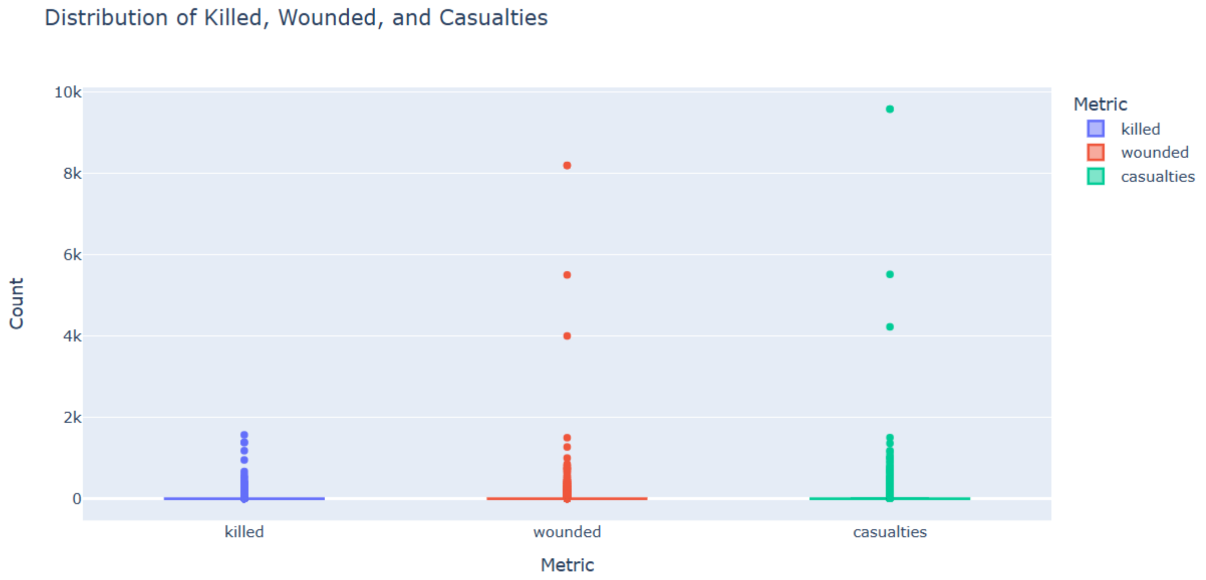
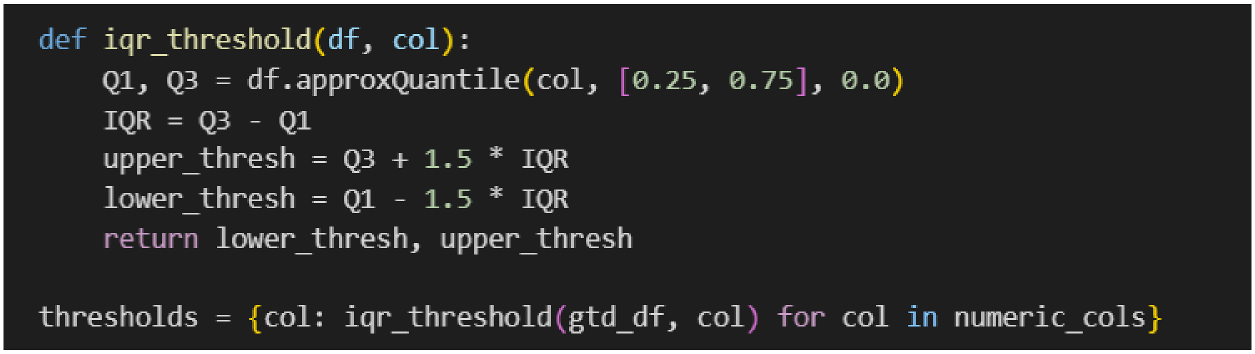


Figure 16 - Boxplot of Distribution of Killed, Wounded, and Casualties

To mitigate their impact, the Interquartile Range (IQR) method was applied. The first (Q1) and third quartiles (Q3) were computed for each variable to calculate the IQR, and upper thresholds were set at Q3 + 1.5*IQR. Rather than removing the identified outliers, a capping approach was used, replacing extreme values beyond the upper bound with the threshold value. This strategy preserved the data structure and sample size while limiting the influence of anomalously high values, particularly in terrorism-related incidents where some cases may report exceptionally large numbers of casualties.



```
def iqr_threshold(df, col):
    Q1, Q3 = df.approxQuantile(col, [0.25, 0.75], 0.0)
    IQR = Q3 - Q1
    upper_thresh = Q3 + 1.5 * IQR
    lower_thresh = Q1 - 1.5 * IQR
    return lower_thresh, upper_thresh

thresholds = {col: iqr_threshold(gtd_df, col) for col in numeric_cols}
```

Figure 17 - Handling Outliers

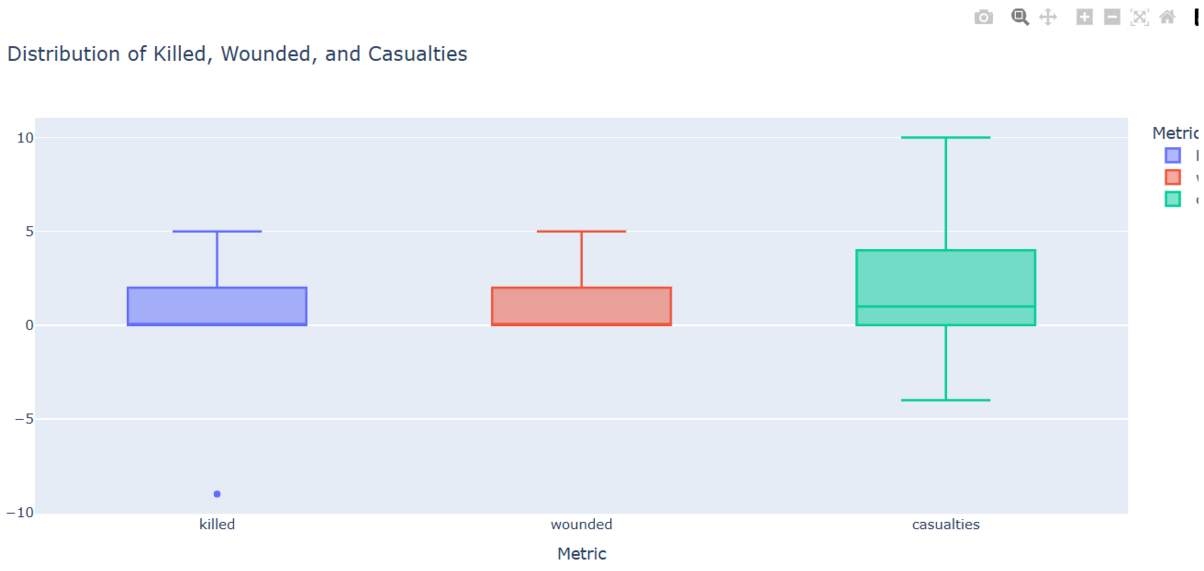
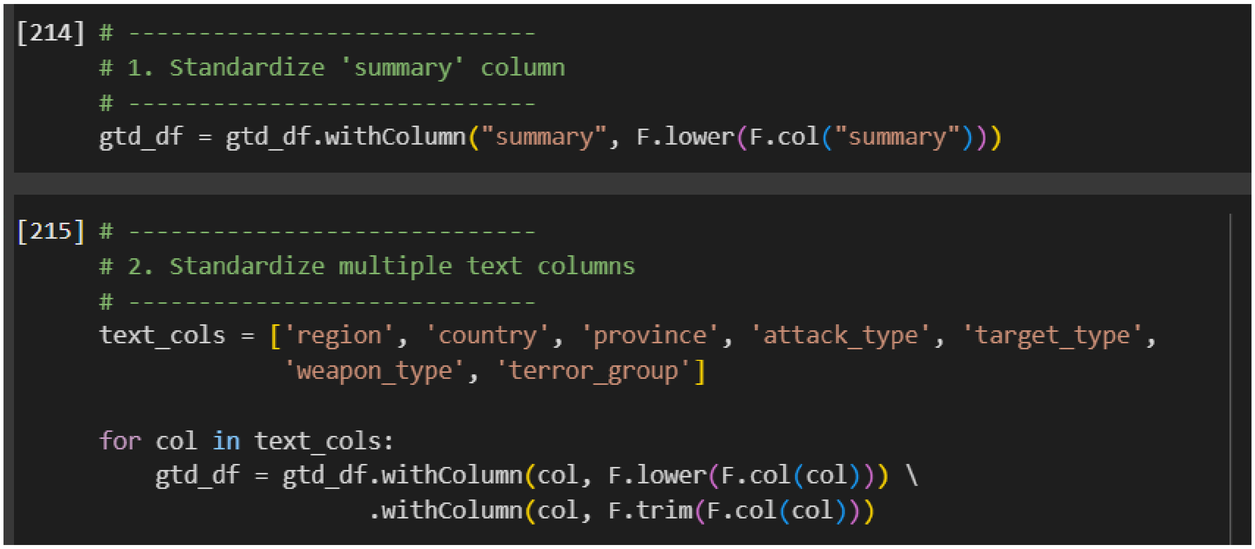


Figure 18 - Box Plot after removing Outliers

Preprocessing for Categorical Variables

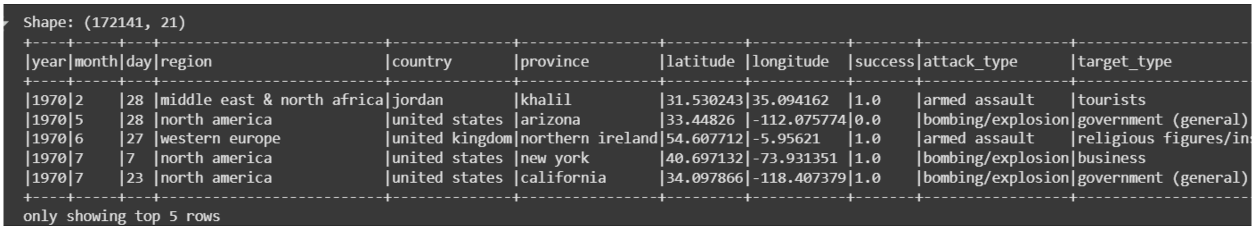
Then, I performed preprocessing for categorical variables to enhance consistency and prepare the data for analysis. Text fields often contain inconsistencies due to varying cases, spacing, or formatting, which can lead to redundant categories and reduce model performance. To address this, the summary column was standardized by converting all text to lowercase using the str.lower() function. Additionally, for key categorical columns such as region, country, province, attack_type, target_type, weapon_type, and terror_group, I applied both str.lower() and str.strip() to ensure that values are case-insensitive and free from leading or trailing whitespaces. This step was essential to avoid duplicate labels and to maintain uniformity across the dataset, thereby improving the reliability of downstream analyses like encoding, grouping, and classification.



```
[214] # ----------------------------
      # 1. Standardize 'summary' column
      # ----------------------------
      gtd_df = gtd_df.withColumn("summary", F.lower(F.col("summary")))

[215] # ----------------------------
      # 2. Standardize multiple text columns
      # ----------------------------
      text_cols = ['region', 'country', 'province', 'attack_type', 'target_type',
                   'weapon_type', 'terror_group']

      for col in text_cols:
          gtd_df = gtd_df.withColumn(col, F.lower(F.col(col))) \
                         .withColumn(col, F.trim(F.col(col)))
```

Figure 19 - Preprocessing for Categorical Variables

Table 4 - Preprocessed Dataset

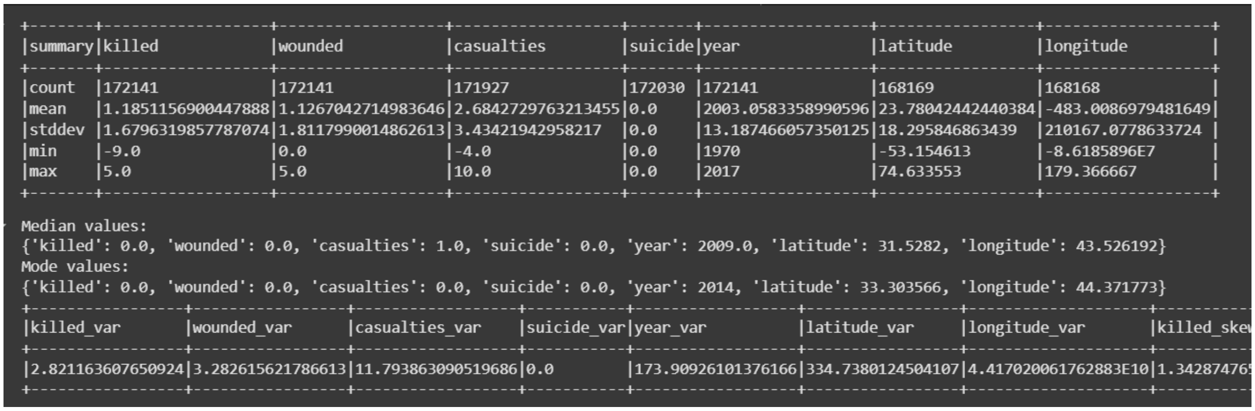


```
Shape: (172141, 21)
+----+-----+---+--------------------------+--------------+----------------+---------+-----------+-------+-----------------+--------------------
|year|month|day|region                    |country       |province        |latitude |longitude  |success|attack_type      |target_type
+----+-----+---+--------------------------+--------------+----------------+---------+-----------+-------+-----------------+--------------------
|1970|2    |28 |middle east & north africa|jordan        |khalil          |31.530243|35.094162  |1.0    |armed assault    |tourists
|1970|5    |28 |north america             |united states |arizona         |33.44826 |-112.075774|0.0    |bombing/explosion|government (general)
|1970|6    |27 |western europe            |united kingdom|northern ireland|54.607712|-5.95621   |1.0    |armed assault    |religious figures/in
|1970|7    |7  |north america             |united states |new york        |40.697132|-73.931351 |1.0    |bombing/explosion|business
|1970|7    |23 |north america             |united states |california      |34.097866|-118.407379|1.0    |bombing/explosion|government (general)
+----+-----+---+--------------------------+--------------+----------------+---------+-----------+-------+-----------------+--------------------
only showing top 5 rows
```

07. Phase 3: Descriptive Analysis

Summary Statistics

To begin the descriptive analysis, I performed summary statistics on key numerical variables, including killed, wounded, casualties, suicide, year, latitude, and longitude. The initial descriptive table provided insights into the central tendency and spread of the data. For instance, both killed and wounded had a mean slightly above 1, with medians of 0, indicating a highly right-skewed distribution, where most incidents result in no casualties, but a few extreme cases contribute to higher averages. The variables *Killed*, *Wounded*, and *Casualties* have mean values of 1.19, 1.13, and 2.68 respectively, with standard deviations of 1.68, 1.81, and 3.43, indicating that while most incidents involve a small number of victims, there is variability in the impact of attacks. The suicide variable also reflected a strong skew, with a mean of 0.038, showing that suicide attacks are relatively rare. Variance and standard deviation metrics showed a considerable spread in casualty figures. Further statistical measures like mode, skewness, and kurtosis confirmed the presence of asymmetry and heavy tails in variables such as suicide and longitude.

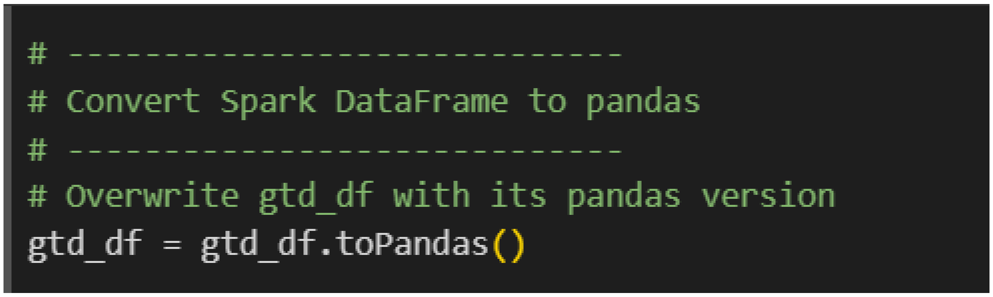


```
+-------+------------------+------------------+------------------+-------+------------------+-----------------+------------------+
|summary|killed            |wounded           |casualties        |suicide|year              |latitude         |longitude         |
+-------+------------------+------------------+------------------+-------+------------------+-----------------+------------------+
|count  |172141            |172141            |171927            |172030 |172141            |168169           |168168            |
|mean   |1.1851156900447888|1.1267042714983646|2.6842729763213455|0.0    |2003.0583358990596|23.78042442440384|-483.0086979481649|
|stddev |1.6796319857787074|1.8117990014862613|3.43421942958217  |0.0    |13.187466057350125|18.295846863439  |210167.0778633724 |
|min    |-9.0              |0.0               |-4.0              |0.0    |1970              |-53.154613       |-8.6185896E7      |
|max    |5.0               |5.0               |10.0              |0.0    |2017              |74.633553        |179.366667        |
+-------+------------------+------------------+------------------+-------+------------------+-----------------+------------------+

Median values:
{'killed': 0.0, 'wounded': 0.0, 'casualties': 1.0, 'suicide': 0.0, 'year': 2009.0, 'latitude': 31.5282, 'longitude': 43.526192}
Mode values:
{'killed': 0.0, 'wounded': 0.0, 'casualties': 0.0, 'suicide': 0.0, 'year': 2014, 'latitude': 33.303566, 'longitude': 44.371773}
+-----------------+-----------------+------------------+-----------+------------------+-----------------+-------------------+---------
|killed_var       |wounded_var      |casualties_var    |suicide_var|year_var          |latitude_var     |longitude_var      |killed_ske
+-----------------+-----------------+------------------+-----------+------------------+-----------------+-------------------+---------
|2.821163607650924|3.282615621786613|11.793863090519686|0.0        |173.90926101376166|334.7380124504107|4.417020061762883E10|1.34287476
+-----------------+-----------------+------------------+-----------+------------------+-----------------+-------------------+---------
```

Figure 20 - Summary Statistics

Exploratory Data Analysis and Visualizations

Then I performed a comprehensive Exploratory Data Analysis.

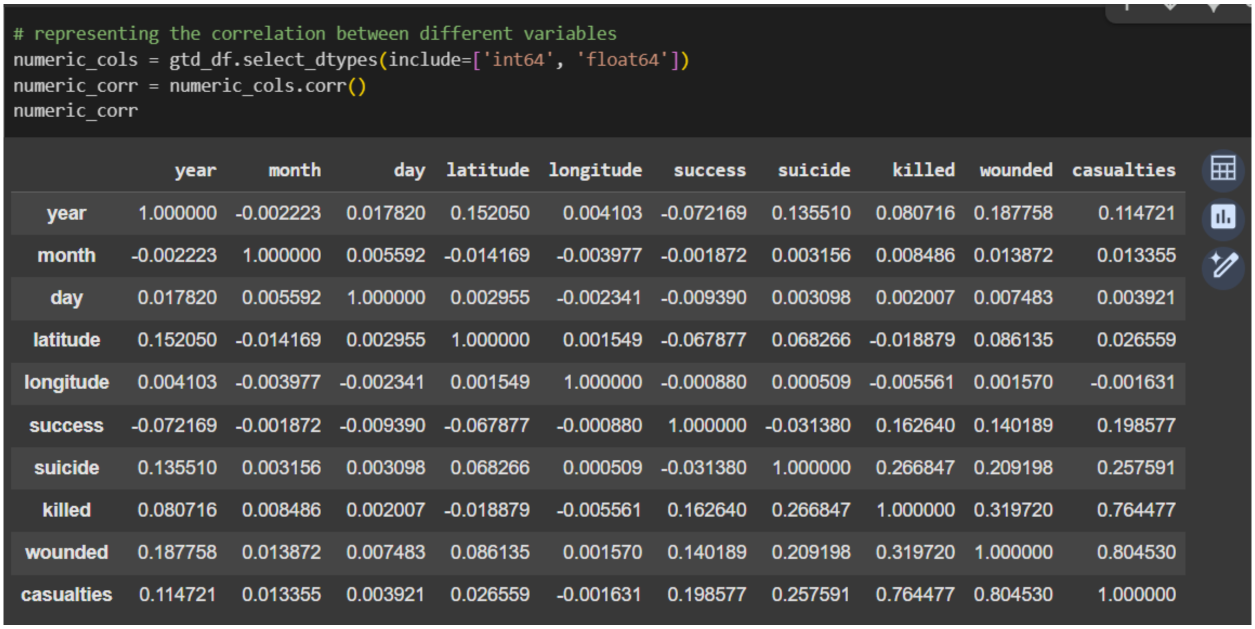


```
# -----------------------------
# Convert Spark DataFrame to pandas
# -----------------------------
# Overwrite gtd_df with its pandas version
gtd_df = gtd_df.toPandas()
```

Figure 21 - Converting to Pandas for EDA

For the exploratory data analysis (EDA) part, I am using pandas instead of Spark because I want to leverage Plotly for visualizations. While Spark is excellent for processing and analyzing large distributed datasets efficiently, its native visualization capabilities are limited. Plotly, on the other hand, provides interactive and highly customizable plots, which are very useful during EDA to explore distributions, trends, correlations, and outliers. By converting the Spark DataFrame to a pandas DataFrame, I can take advantage of Plotly's interactive charts while still working with the cleaned and preprocessed data. This approach combines the scalability of Spark for data processing with the visual power of Plotly for insights.

For the Exploratory Data Analysis (EDA), the first step involved examining the correlation matrix among key numerical variables such as year, month, day, latitude, longitude, success, suicide, killed, wounded, and casualties. The correlation analysis revealed several noteworthy relationships: for instance, killed and wounded showed a strong positive correlation with casualties (0.764 and 0.805, respectively), as expected due to their additive nature. Additionally, suicide incidents were positively correlated with both killed (0.267) and wounded (0.209), indicating that suicide attacks tend to result in higher casualties. There were also moderate correlations between success and variables like killed (0.163), wounded (0.140), and casualties (0.199), suggesting that successful attacks generally lead to greater human impact. On the other hand, temporal variables such as year, month, and day showed negligible correlation with attack severity, implying that casualties are not directly associated with the calendar date. Overall, the correlation matrix helped identify variables with strong linear relationships, offering valuable insights for feature selection and further analysis.



```
# representing the correlation between different variables
numeric_cols = gtd_df.select_dtypes(include=['int64', 'float64'])
numeric_corr = numeric_cols.corr()
numeric_corr
```

	year	month	day	latitude	longitude	success	suicide	killed	wounded	casualties
year	1.000000	-0.002223	0.017820	0.152050	0.004103	-0.072169	0.135510	0.080716	0.187758	0.114721
month	-0.002223	1.000000	0.005592	-0.014169	-0.003977	-0.001872	0.003156	0.008486	0.013872	0.013355
day	0.017820	0.005592	1.000000	0.002955	-0.002341	-0.009390	0.003098	0.002007	0.007483	0.003921
latitude	0.152050	-0.014169	0.002955	1.000000	0.001549	-0.067877	0.068266	-0.018879	0.086135	0.026559
longitude	0.004103	-0.003977	-0.002341	0.001549	1.000000	-0.000880	0.000509	-0.005561	0.001570	-0.001631
success	-0.072169	-0.001872	-0.009390	-0.067877	-0.000880	1.000000	-0.031380	0.162640	0.140189	0.198577
suicide	0.135510	0.003156	0.003098	0.068266	0.000509	-0.031380	1.000000	0.266847	0.209198	0.257591
killed	0.080716	0.008486	0.002007	-0.018879	-0.005561	0.162640	0.266847	1.000000	0.319720	0.764477
wounded	0.187758	0.013872	0.007483	0.086135	0.001570	0.140189	0.209198	0.319720	1.000000	0.804530
casualties	0.114721	0.013355	0.003921	0.026559	-0.001631	0.198577	0.257591	0.764477	0.804530	1.000000

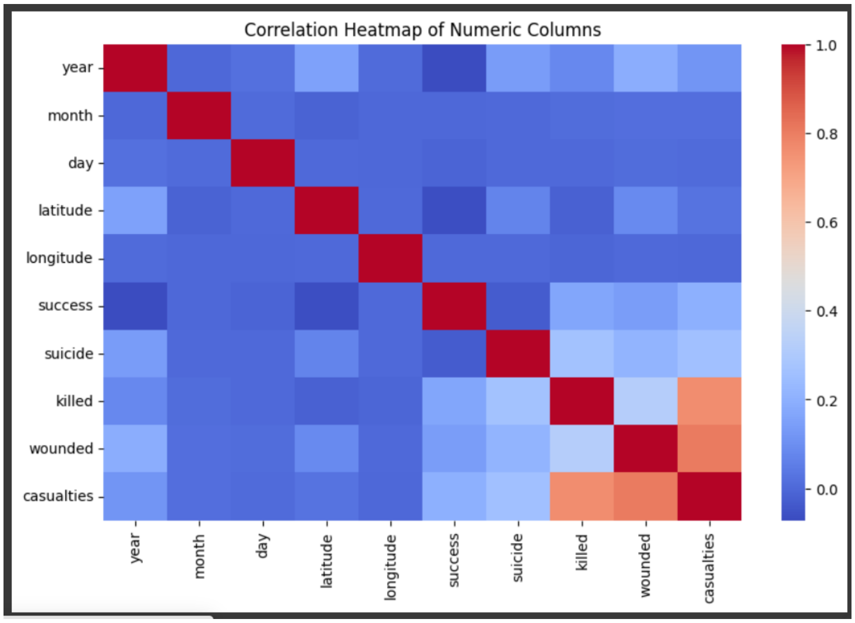


Figure 22 - Correlation Matrix

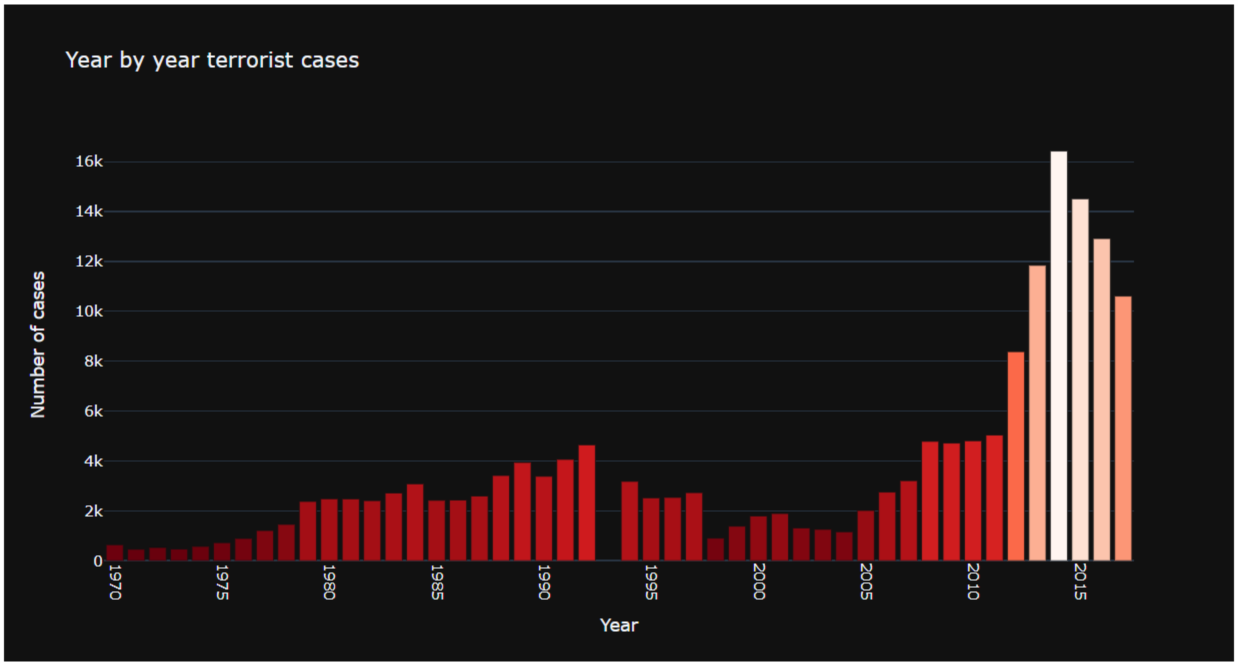


Figure 23 - Year by year terrorist cases

The first bar plot shows the number of terrorist incidents recorded each year. From the early 1970s to around 2004, the number of cases remained relatively low and stable. However, post-2004, there is a sharp increase in terrorism incidents, peaking between 2014 and 2015. This trend highlights a significant rise in global terrorist activities during the 2010s.

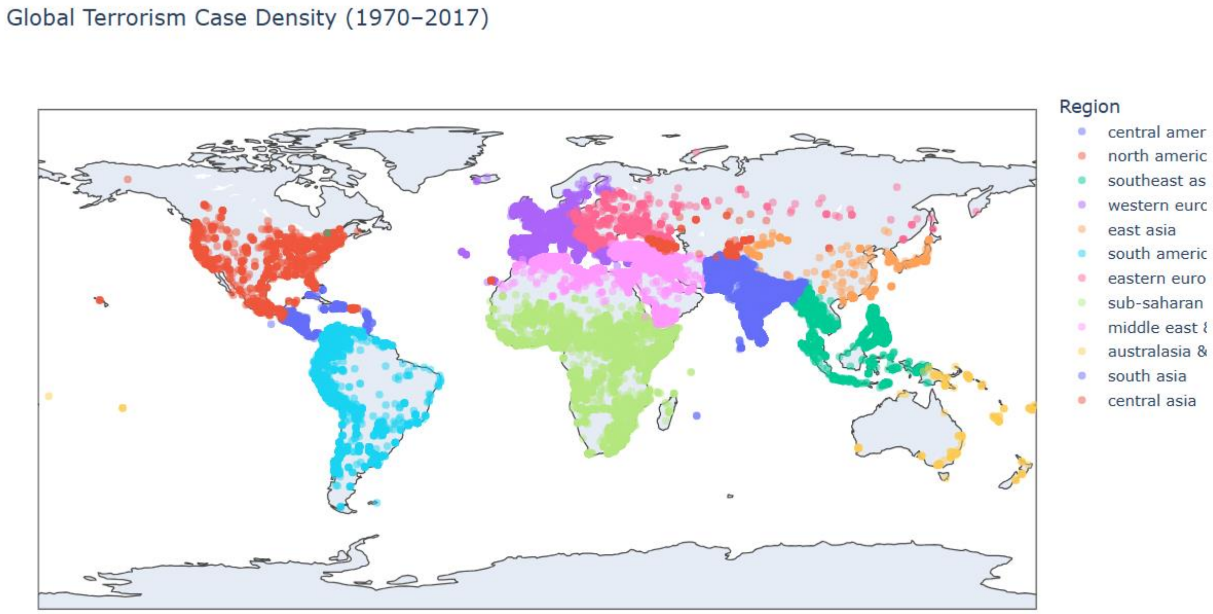


Figure 24 - Global Terrorism Case Density

The plot "Global Terrorism Case Density (1970–2017)" visually represents the concentration of terrorist incidents across different regions over a 47-year period. It highlights significant geographical disparities, with the Middle East & North Africa (MENA) and South Asia appearing as the darkest or most densely marked areas, indicating the highest frequency of attacks. Regions like Sub-Saharan Africa, Southeast Asia, and Central Asia show moderate density, while North America, Western Europe, and Australasia appear sparsely marked, reflecting lower terrorism prevalence. The visualization effectively underscores how terrorism is heavily concentrated in conflict-prone regions, with MENA and South Asia bearing the brunt of attacks, while more stable regions experience comparatively minimal activity.

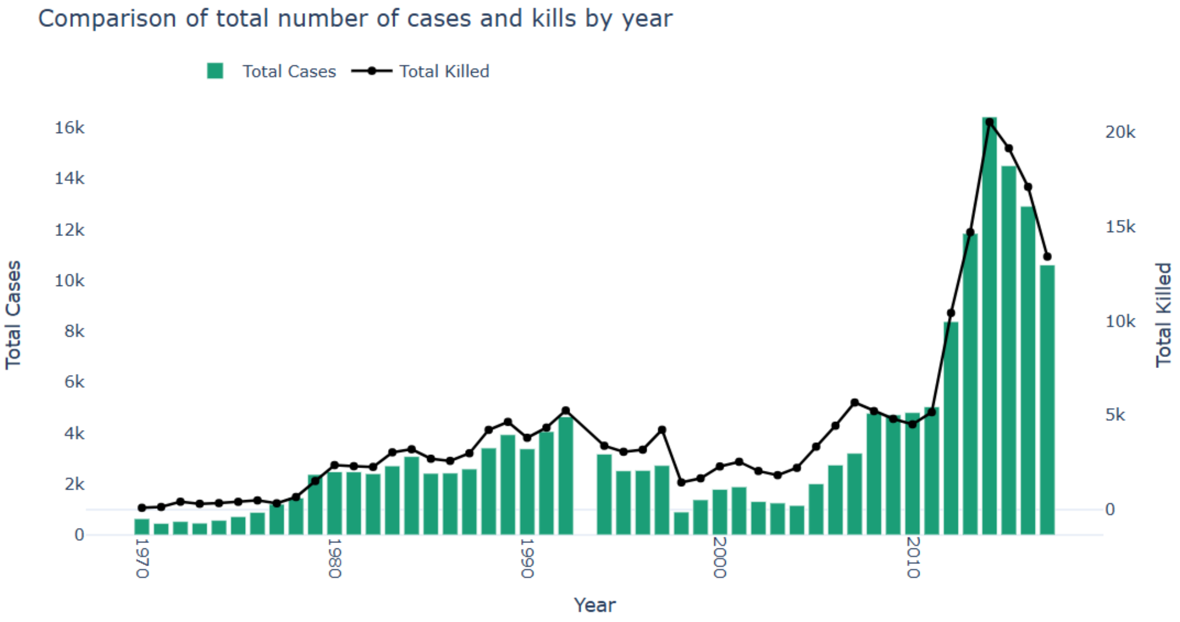


Figure 25 - Comparison of total number of cases and kills by year

This plot combines bar and line graphs to compare the total number of terrorism-related incidents and total fatalities over time. While the number of cases increased sharply after 2004, the number of deaths also rose dramatically, especially around 2014, suggesting not only more frequent attacks but also deadlier ones during that period. This helps underline the growing impact and severity of terrorism in recent years.

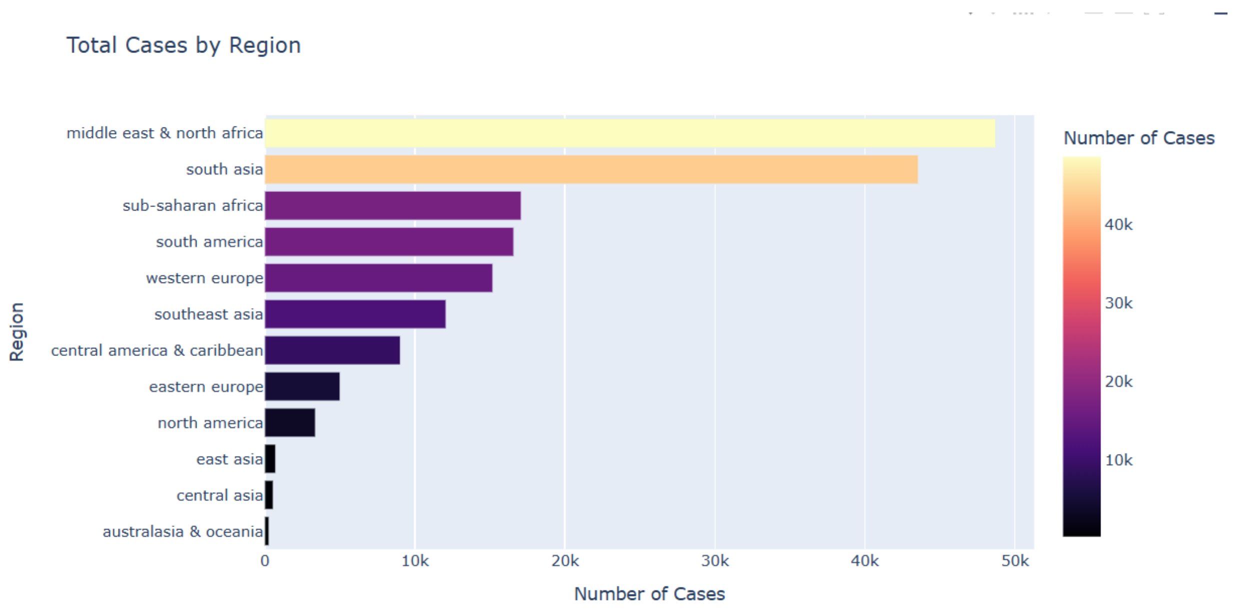


Figure 26 - Total Cases by Region

The third horizontal bar chart visualizes the regional distribution of terrorist cases. The Middle East & North Africa and South Asia are shown to be the most affected regions, each with over 40,000 recorded cases. Sub-Saharan Africa and South America follow with significantly lower but still notable numbers. This regional breakdown highlights how certain parts of the world are disproportionately impacted by terrorism.

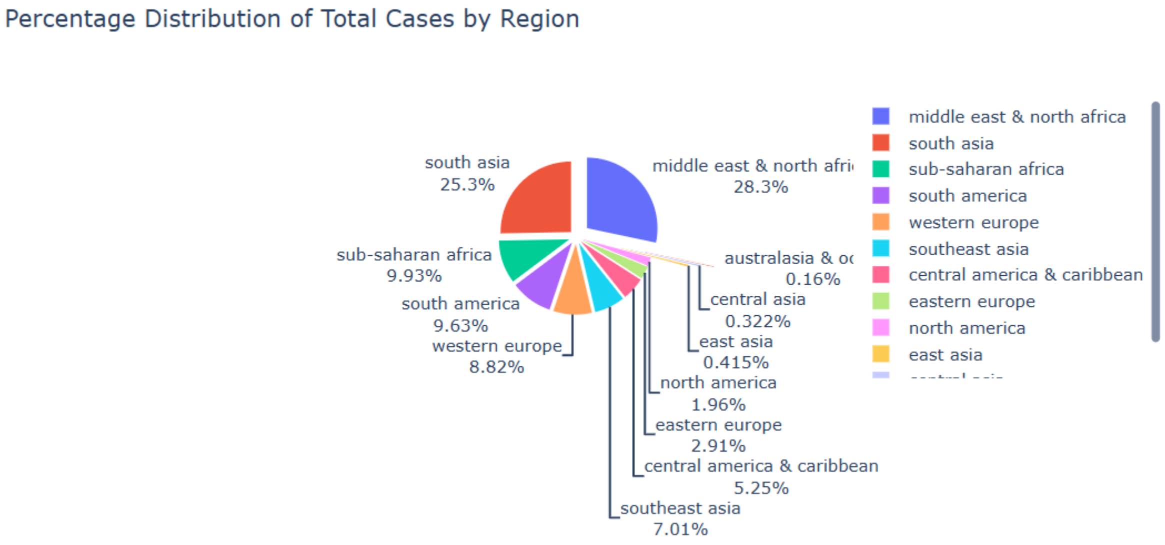


Figure 27 - Percentage Distribution of Total Cases by Region

The pie chart # illustrates how terrorist incidents are distributed across different regions. The Middle East & North Africa account for the highest proportion at 28.3%, followed closely by South Asia at 25.3%. Sub-Saharan Africa, South America, and Western Europe also contribute significantly, with percentages ranging from 8.82% to 9.93%. In contrast, regions like Australasia & Oceania, Central Asia, and East Asia have minimal contributions, each below 1%. This distribution highlights the concentration of terrorist activities in specific geopolitical areas, likely influenced by factors such as political instability, conflict, and socio-economic conditions.

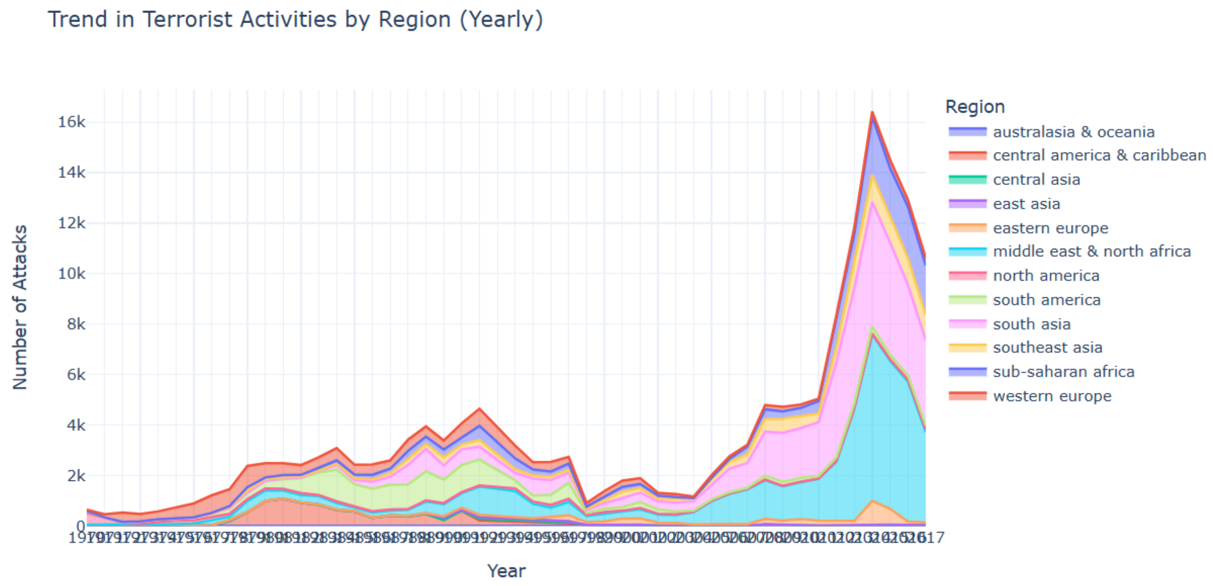


Figure 28 - Trend in Terrorist Activities by Region (Yearly)

The graph showing the "Trend in Terrorist Activities by Region (Yearly)" depicts the yearly number of terrorist attacks across regions over time. Regions like the Middle East & North Africa and South Asia probably show higher numbers, consistent with their dominance in the percentage distribution. The trend may indicate periods of escalation or decline, possibly correlating with geopolitical events, counter-terrorism efforts, or regional conflicts. Such trends are critical for understanding the dynamic nature of terrorism globally.

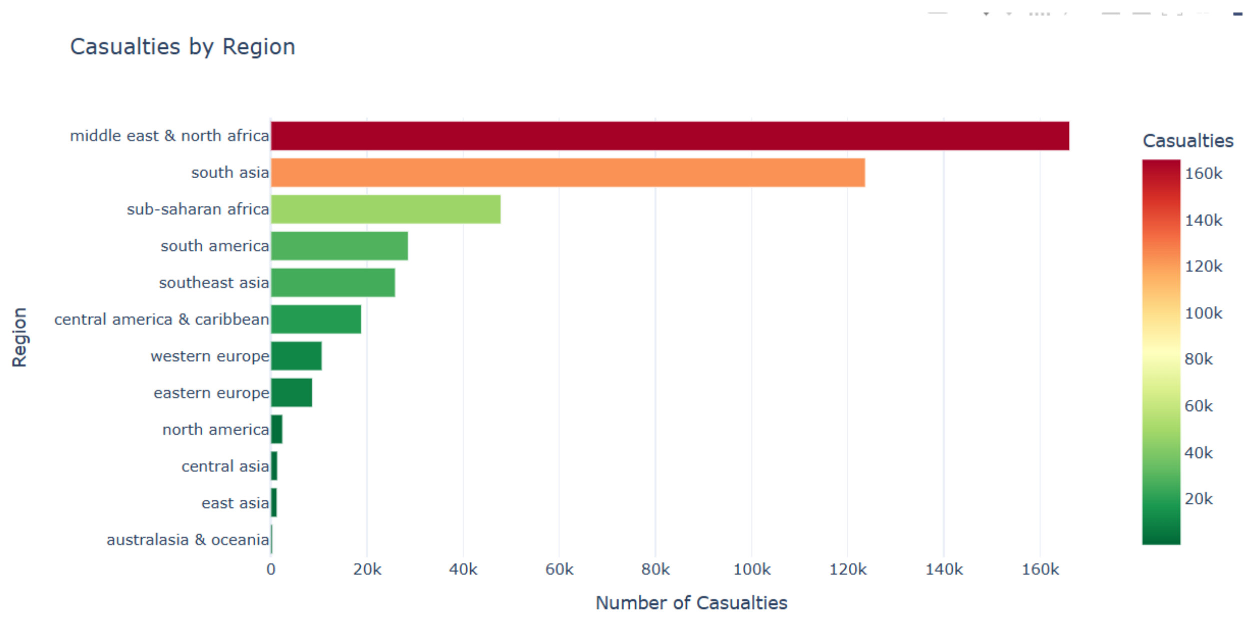


Figure 29 - Casualties by Region

The bar graph "Casualties by Region" ranks regions by the total number of casualties (deaths and injuries) caused by terrorist attacks. The Middle East & North Africa and South Asia likely lead with the highest casualties, reflecting their high attack frequencies and the severity of incidents. Sub-Saharan Africa and South America #follow, with Southeast Asia and Central America & the Caribbean also contributing notable numbers. Western Europe, Eastern Europe, and North America, while having fewer casualties, still show measurable impacts. This graph underscores the human cost of terrorism, with the most affected regions bearing the brunt of fatalities and injuries, often due to large-scale or sustained conflicts.

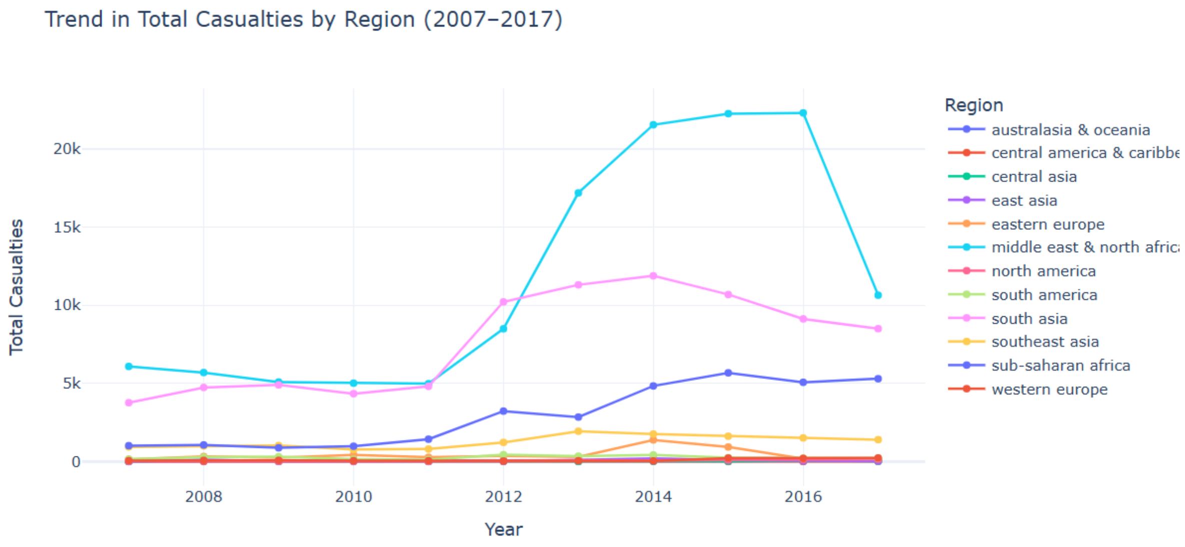


Figure 30 - Trend in Total Casualties by Region (2007–2017)

This line graph tracks the total casualties (deaths and injuries) from terrorist attacks across different regions from 2007 to 2017. The Middle East & North Africa and South Asia likely show the highest peaks, reflecting ongoing conflicts and instability. Sub-Saharan Africa and South America may also exhibit rising trends, while regions like Western Europe and North America remain relatively low but may show sporadic spikes due to isolated high-casualty attacks. The graph helps identify periods of escalation, such as post-2011 during the Arab Spring or the rise of ISIS, and highlights regions requiring urgent counter-terrorism measures.

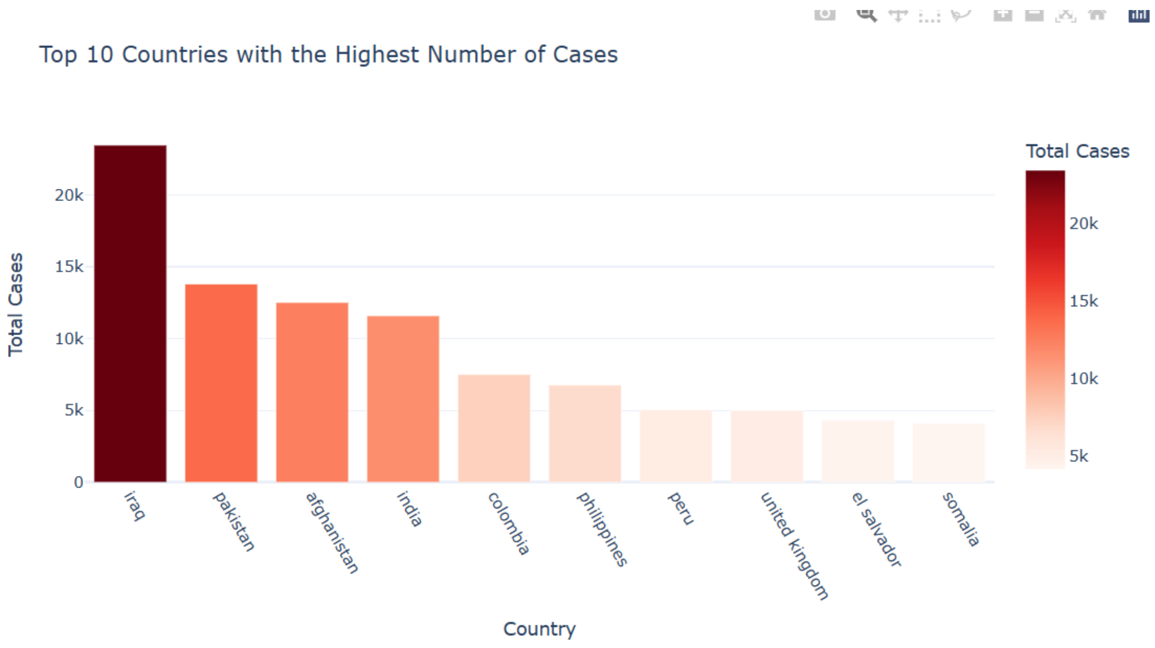


Figure 31 - Top 10 Countries with the Highest Number of Cases

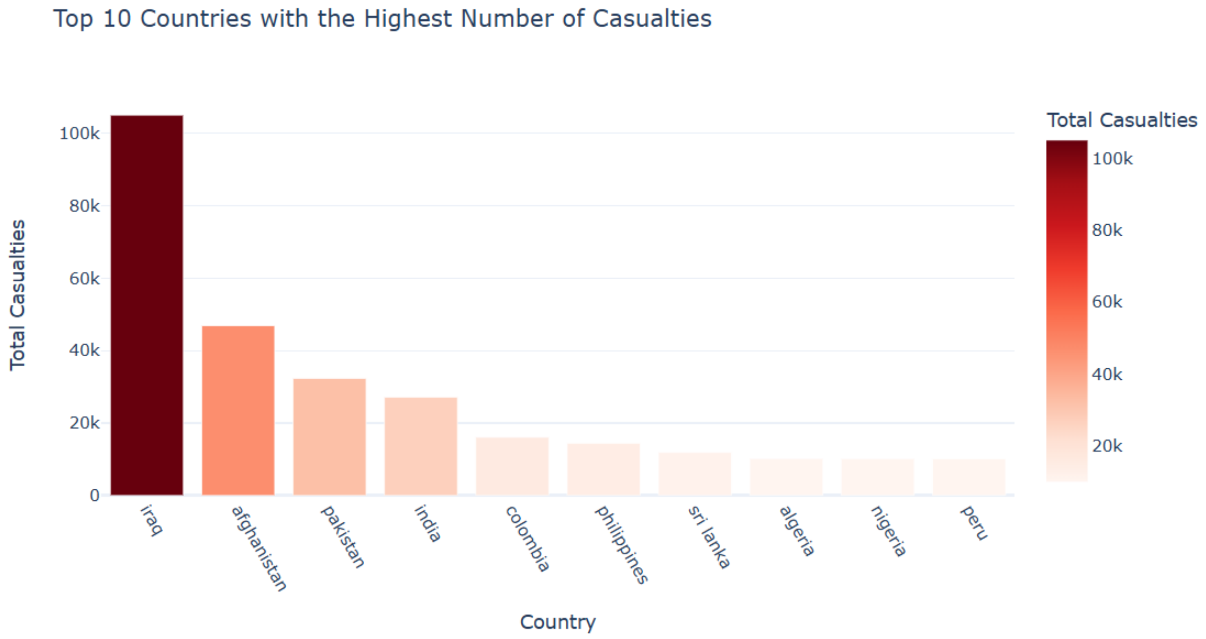


Figure 32 - Top 10 Countries with the Highest Number of Casualties

The above 2 bar charts ranks countries by their total recorded terrorist incidents as well as causalities. Iraq, Afghanistan, and Pakistan likely dominate due to prolonged insurgencies and extremist activity. India, the Philippines, and Nigeria may also appear, linked to separatist movements and jihadist groups. The presence of the UK or other Western nations could reflect domestic extremism or high-profile attacks. The visualization underscores how terrorism is concentrated in specific nations, often tied to governance failures, ethnic conflicts, or foreign interventions.

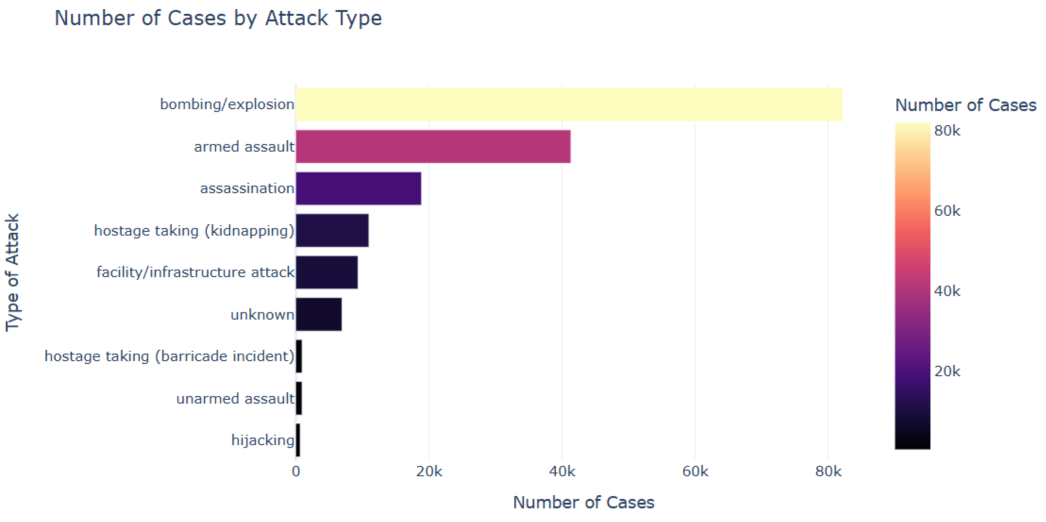


Figure 33 - Number of Cases by Attack Type

This bar graph categorizes terrorist incidents by attack methodology. Bombings/explosions and armed assaults are likely the most frequent, given their lethality and ease of execution. Assassinations and hostage-takings (kidnappings) follow, while hijackings and barricade incidents are rarer. The dominance of bombings aligns with global trends, as they maximize psychological impact and media attention. Understanding these patterns helps security agencies prioritize preventive measures, such as explosive detection or counter-assault training.

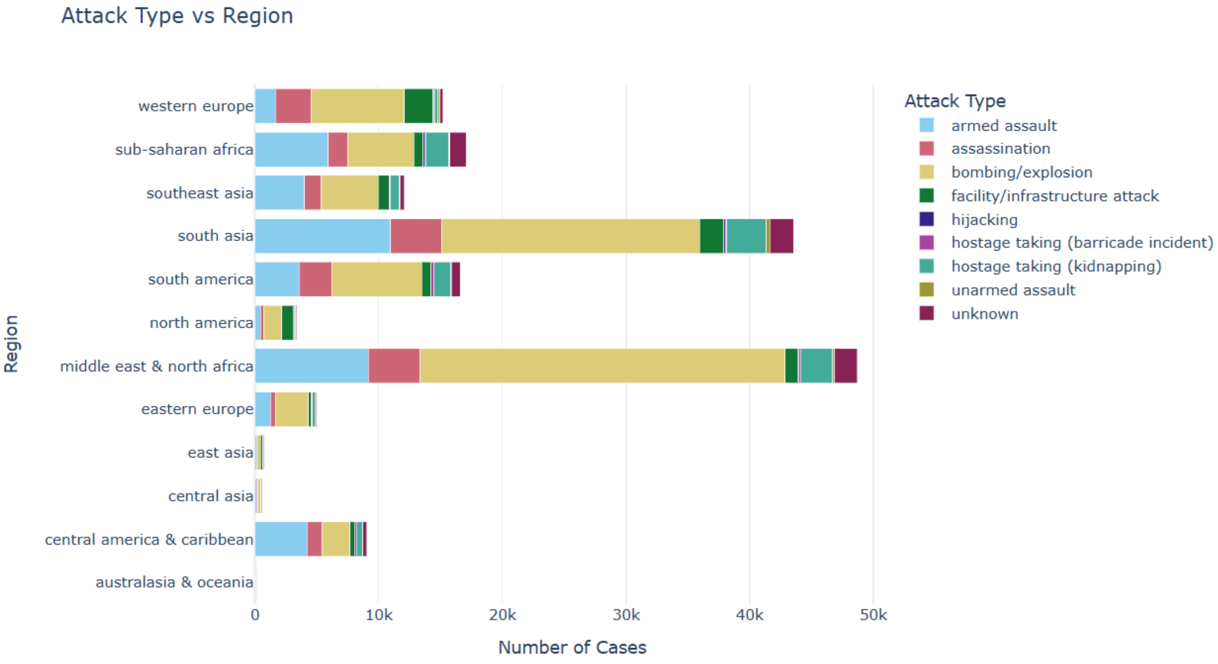


Figure 34 - Attack Type vs Region

This stacked bar chart examines how attack methods vary by region. The Middle East & South Asia may show heavy use of bombings and armed assaults, typical of insurgencies. Sub-Saharan Africa could see more armed assaults and kidnappings, reflecting guerrilla tactics. In contrast, Western Europe and North America may have fewer but more diverse attacks, including unarmed assaults or facility attacks, often tied to lone-actor extremism. This regional breakdown aids in tailoring counter-terrorism strategies for example, bomb-disposal units in conflict zones versus surveillance in urban areas.

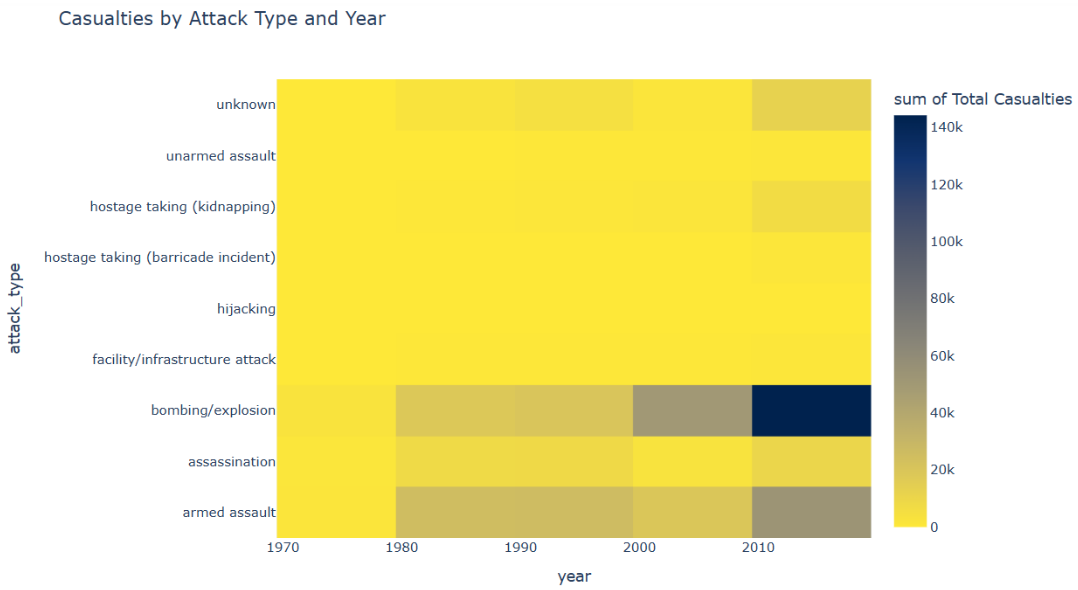


Figure 35 - Casualties by Attack Type and Year

This heatmap visualizes the relationship between attack types and casualties over time (1970–2010). Darker shades indicate higher casualties, revealing trends such as the dominance of bombings/explosions and armed assaults in causing mass casualties, particularly during peak conflict years (e.g., 2000s in Iraq/Afghanistan). Lighter years for tactics like hijackings or assassinations suggest their lower lethality or declining use. The heatmap highlights how specific attack methods drive surges in violence, correlating with geopolitical events like the rise of jihadist groups post-9/11.

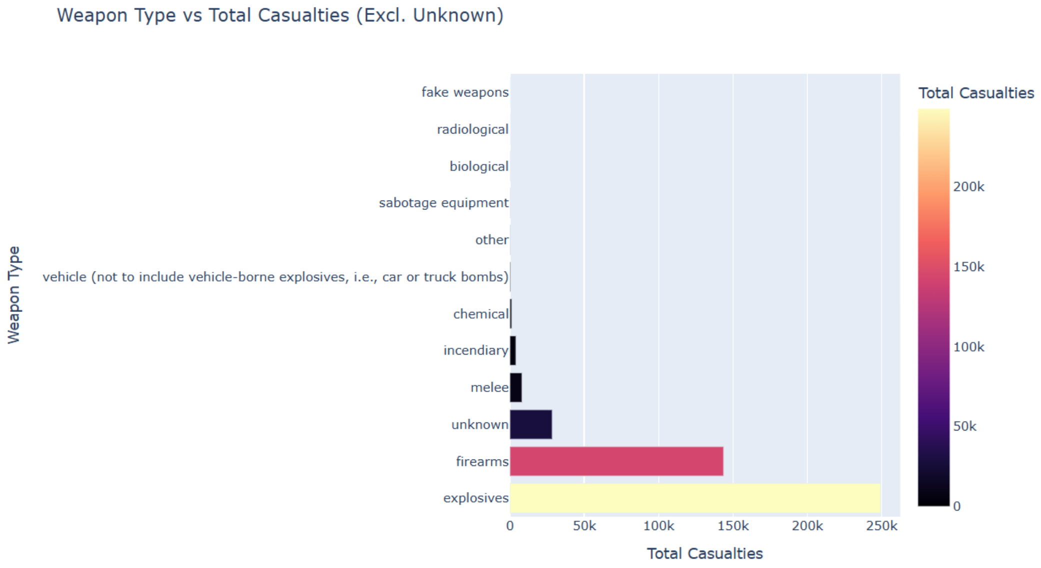


Figure 36 - Weapon Type vs Total Casualties (Excl. Unknown)

This bar chart ranks weapon types by their associated casualties. Explosives and firearms dominate, reflecting their widespread use in attacks like bombings and shootings. Incendiary weapons and melee attacks (e.g., knives) follow, while chemical/biological weapons appear minimal due to their rarity. The stark contrast underscores how conventional weapons remain terrorists' primary tools due to accessibility and destructive potential, guiding security focus on explosive detection and arms control.

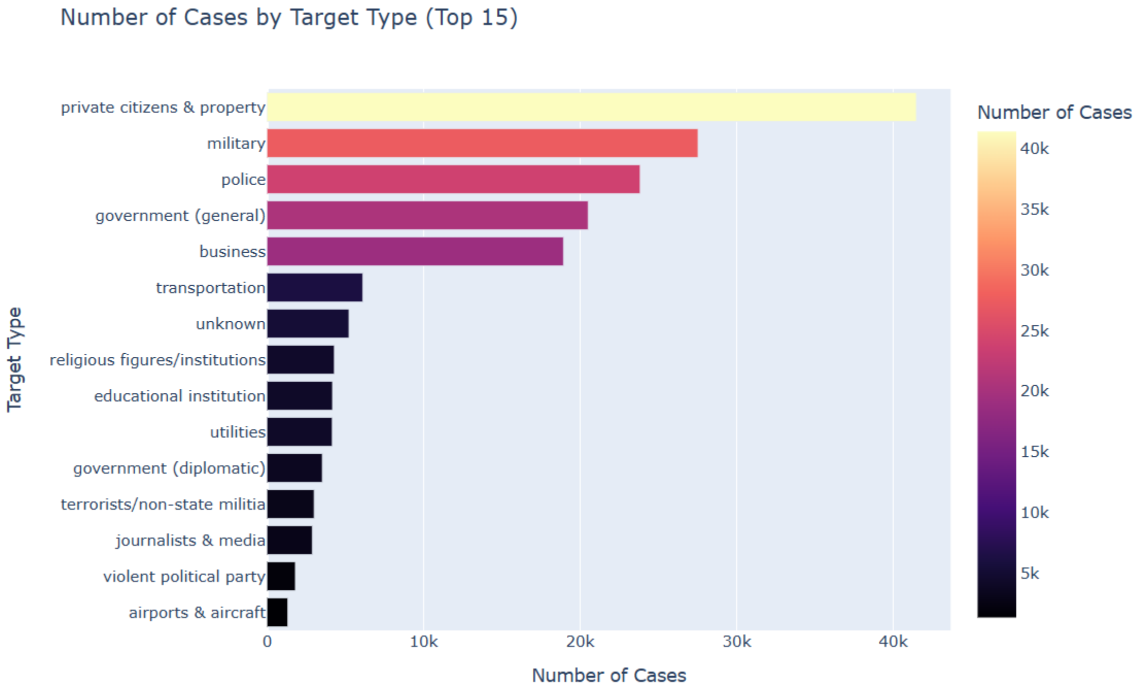


Figure 37 - Number of Cases by Target Type (Top 15)

Private citizens, military, and police are the most frequent targets, emphasizing terrorists' aim to instill fear and challenge state authority. Attacks on transportation (e.g., airports) and religious institutions highlight symbolic value, while utilities and media reflect disruption goals. This breakdown aids in prioritizing protection for high-risk sectors like public spaces and critical infrastructure.

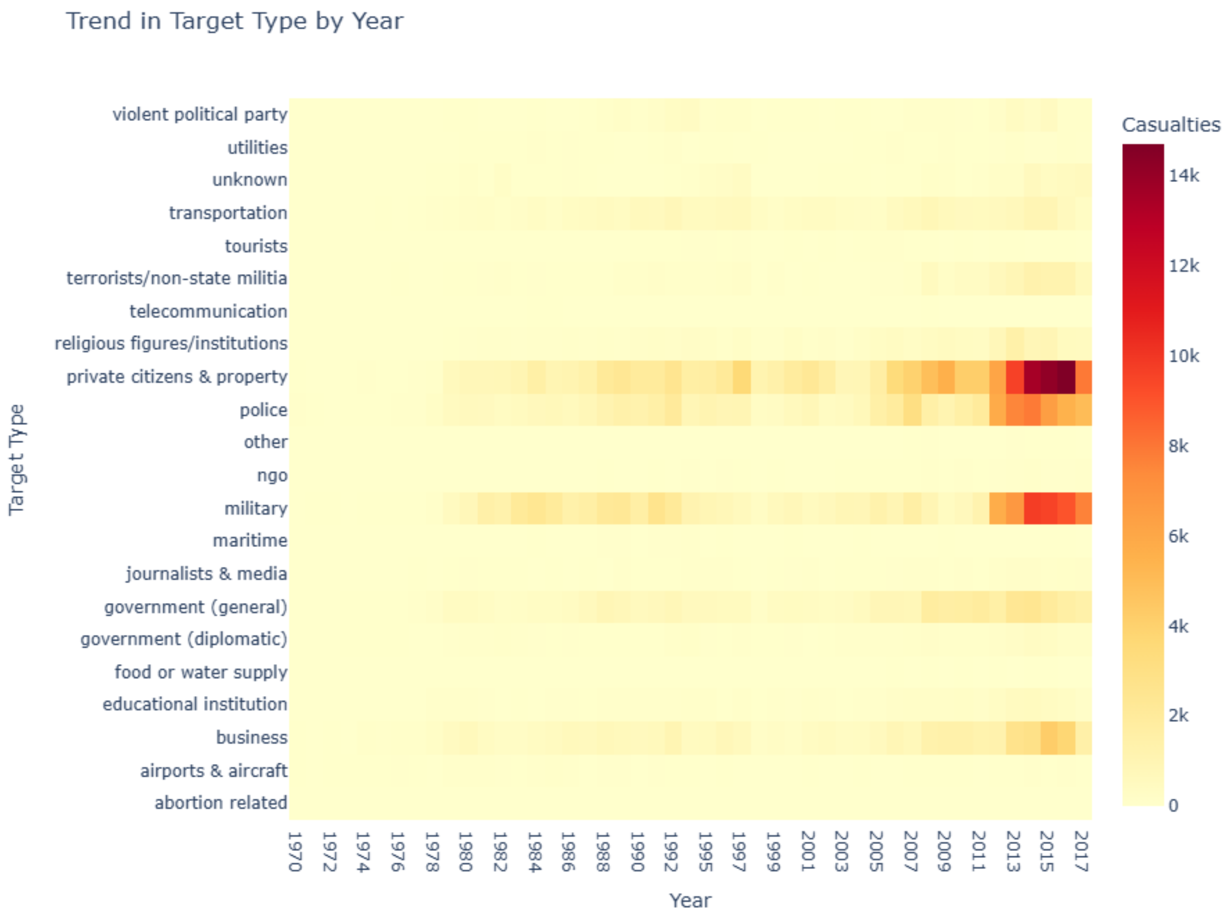


Figure 38 - Trend in Target Type by Year

This heatmap tracks how target preferences evolve annually. Military and police likely show consistent targeting, while spikes in attacks on private citizens or religious sites may align with sectarian violence (e.g., Iraq’s civil war). Shifts toward "soft" targets (e.g., tourists, schools) in later years could indicate counter-terrorism pressures forcing adaption. The trend reveals strategic shifts in terrorist tactics over decades.

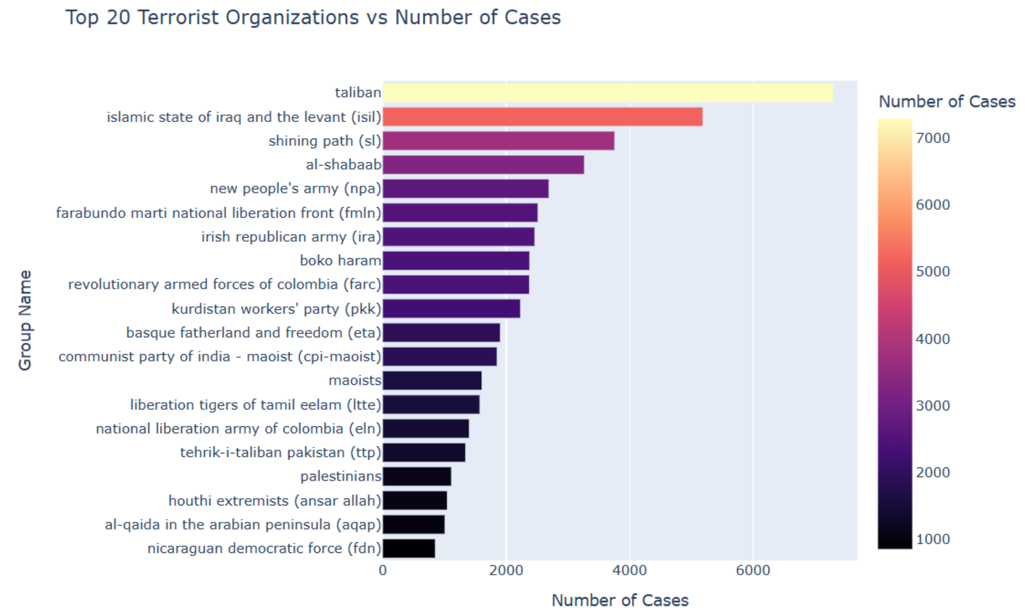


Figure 39- Top 20 Terrorist Organizations vs Number of Cases

The Taliban, ISIS, and Boko Haram lead, reflecting their operational scale in conflict zones like Afghanistan and Nigeria. Leftist groups (e.g., FARC, Shining Path) and separatists (e.g., PKK, LTTE) also appear, tied to historical insurgencies. The data underscores how a few groups drive global terrorism, with ideology (jihadism, communism) and regional grievances shaping their prevalence. This informs counter-terrorism prioritization of high-threat entities.

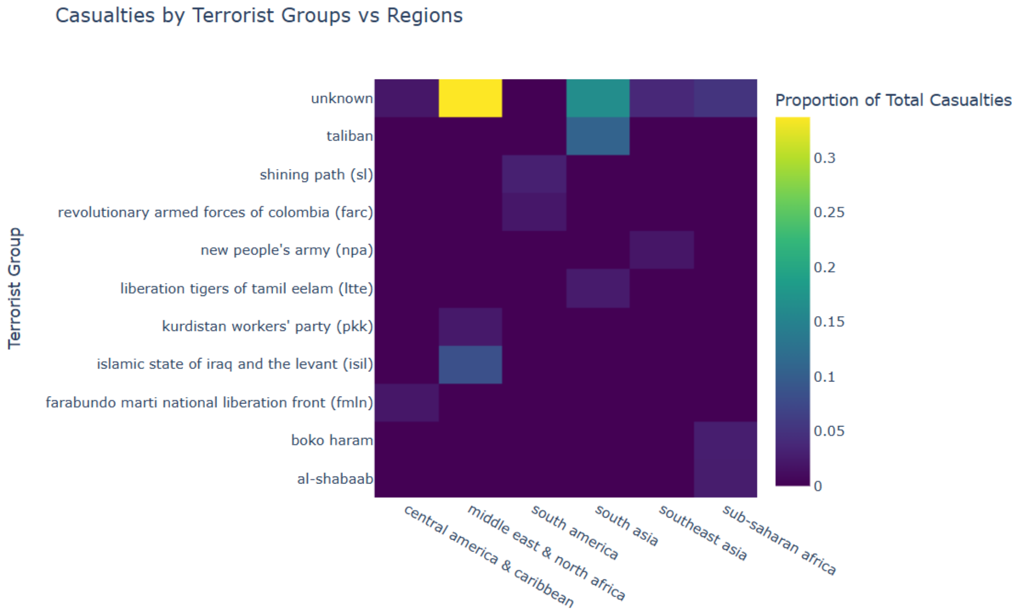


Figure 40 - Casualties by Terrorist Groups vs Regions

This visualization illustrates the disproportionate impact of specific terrorist groups across different regions. The Taliban and ISIS (Islamic State of Iraq and the Levant) dominate in South Asia and the Middle East & North Africa, respectively, accounting for the highest casualties. Boko Haram’s stronghold in Sub-Saharan Africa and the Liberation Tigers of Tamil Eelam’s (LTTE) historical impact in South Asia are also evident. This plot underscores how regional instability and ideological movements shape the operational focus and lethality of terrorist organizations.

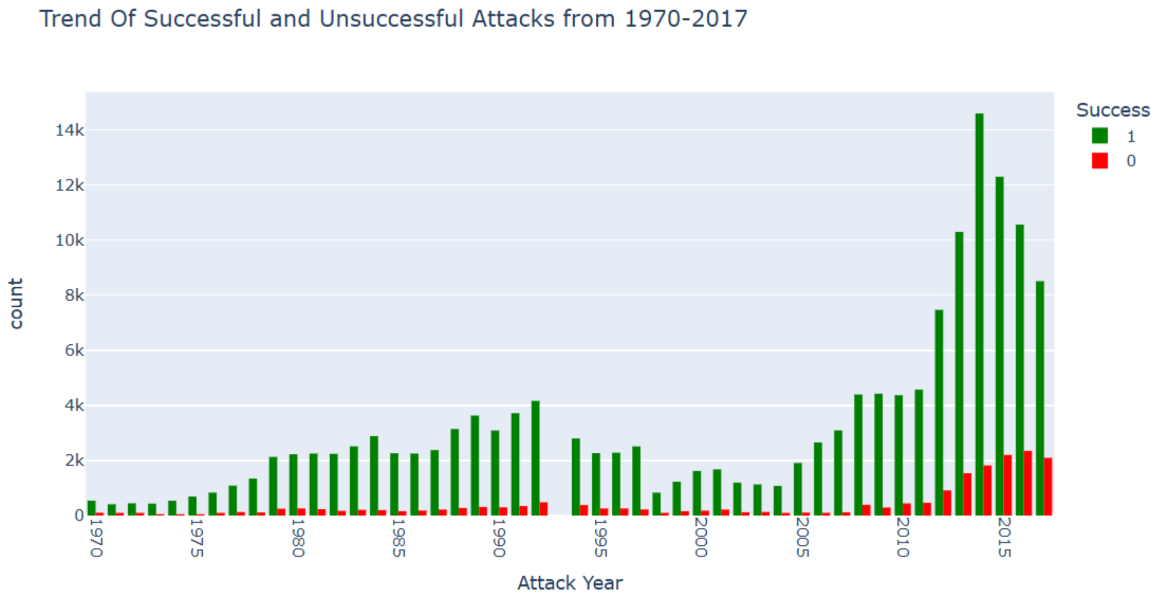


Figure 41 - Trend Of Successful and Unsuccessful Attacks from 1970-2017

The line graph tracks the success rate of terrorist attacks over time, revealing fluctuations linked to counter-terrorism efforts and group capabilities. Peaks in the 1990s and post-2000s correlate with the rise of groups like Al-Qaeda and ISIS, while dips may reflect improved security measures. Unsuccessful attacks (e.g., thwarted plots) are notably lower, emphasizing terrorists’ persistence despite interventions. This trend highlights the evolving "cat-and-mouse" dynamic between terrorists and security forces.

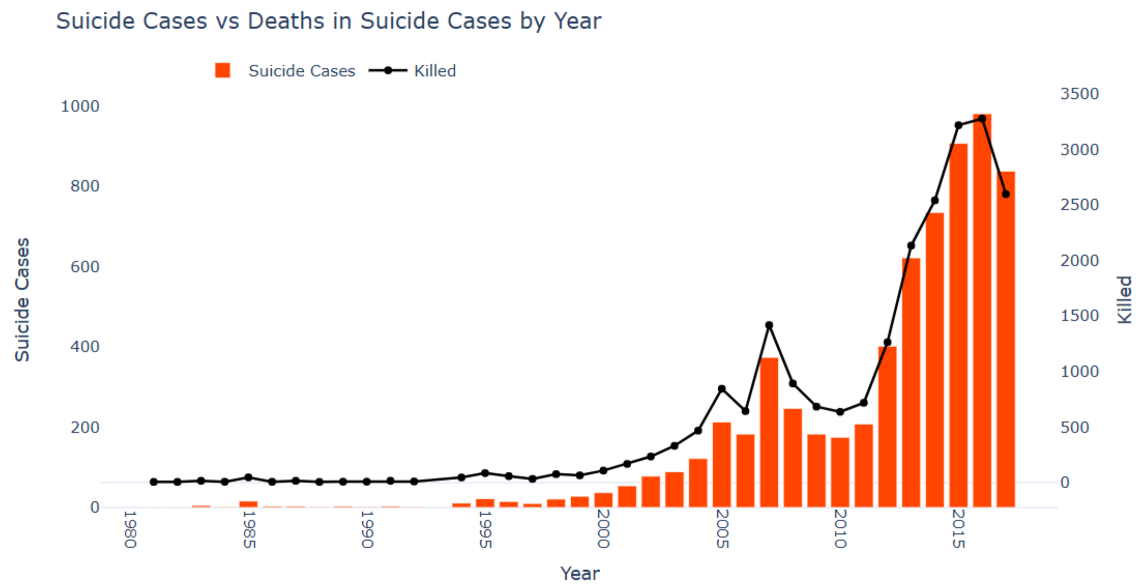
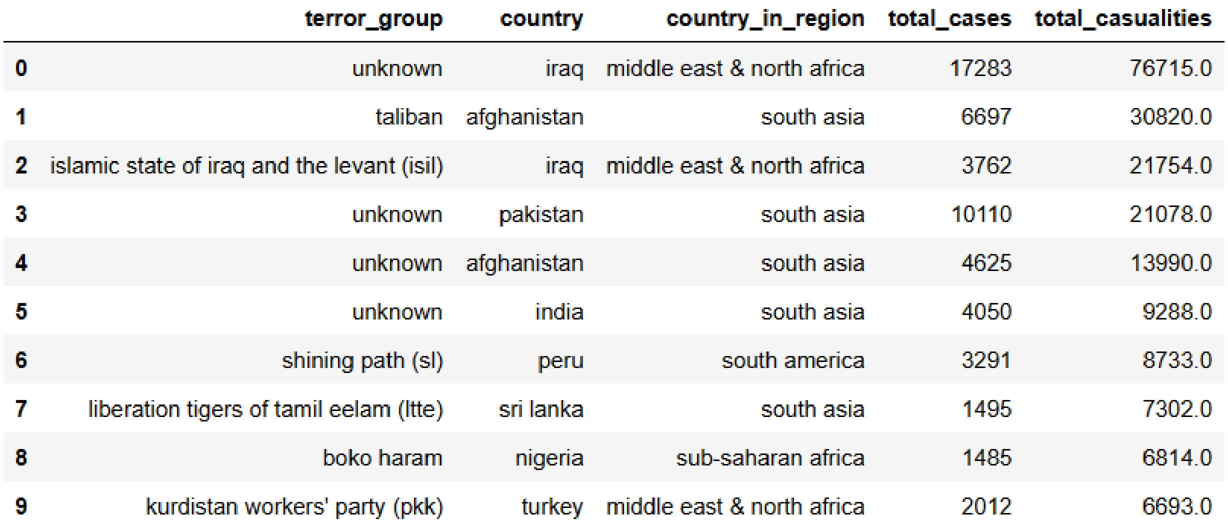


Figure 42 - Suicide Cases vs Deaths in Suicide Cases by Year

This dual-axis chart compares the frequency of suicide attacks to their lethality (deaths per attack). Post-2000, suicide bombings surge, particularly in the Middle East and South Asia, with ISIS and the Taliban maximizing casualties through coordinated strikes. The parallel rise in deaths per attack suggests tactical refinement, such as vehicle-borne explosives targeting crowds. This grim trend underscores suicide terrorism's role as a high-impact strategy for instilling terror.

Table 5 - Top Terrorist Groups by Casualties and Region



	terror_group	country	country_in_region	total_cases	total_casualities
0	unknown	iraq	middle east & north africa	17283	76715.0
1	taliban	afghanistan	south asia	6697	30820.0
2	islamic state of iraq and the levant (isil)	iraq	middle east & north africa	3762	21754.0
3	unknown	pakistan	south asia	10110	21078.0
4	unknown	afghanistan	south asia	4625	13990.0
5	unknown	india	south asia	4050	9288.0
6	shining path (sl)	peru	south america	3291	8733.0
7	liberation tigers of tamil eelam (ltte)	sri lanka	south asia	1495	7302.0
8	boko haram	nigeria	sub-saharan africa	1485	6814.0
9	kurdistan workers' party (pkk)	turkey	middle east & north africa	2012	6693.0

Sri Lanka

My analysis specifically examines Sri Lanka's experience with terrorism, focusing on the Liberation Tigers of Tamil Eelam (LTTE) insurgency and other violent actors. The following visualizations reveal patterns in attacks, casualties, and tactics during decades of conflict:

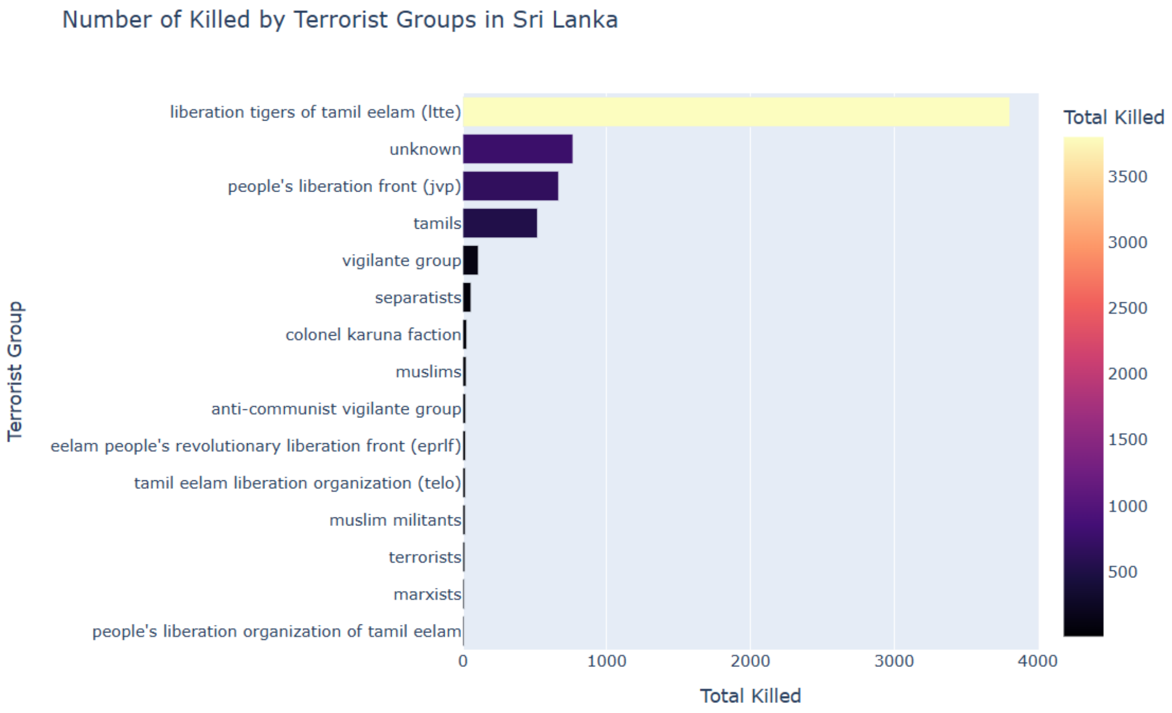


Figure 43 - Number of Killed by Terrorist Groups in Sri Lanka

The bar chart highlights the LTTE as the deadliest group, responsible for thousands of deaths during Sri Lanka's civil war (1983–2009). Factions like the JVP (Marxist insurgents) also contributed significantly. The data reflects the multi-actor nature of Sri Lanka's conflicts, where ethnic strife (Tamil vs. Sinhalese) and ideological movements fueled violence.

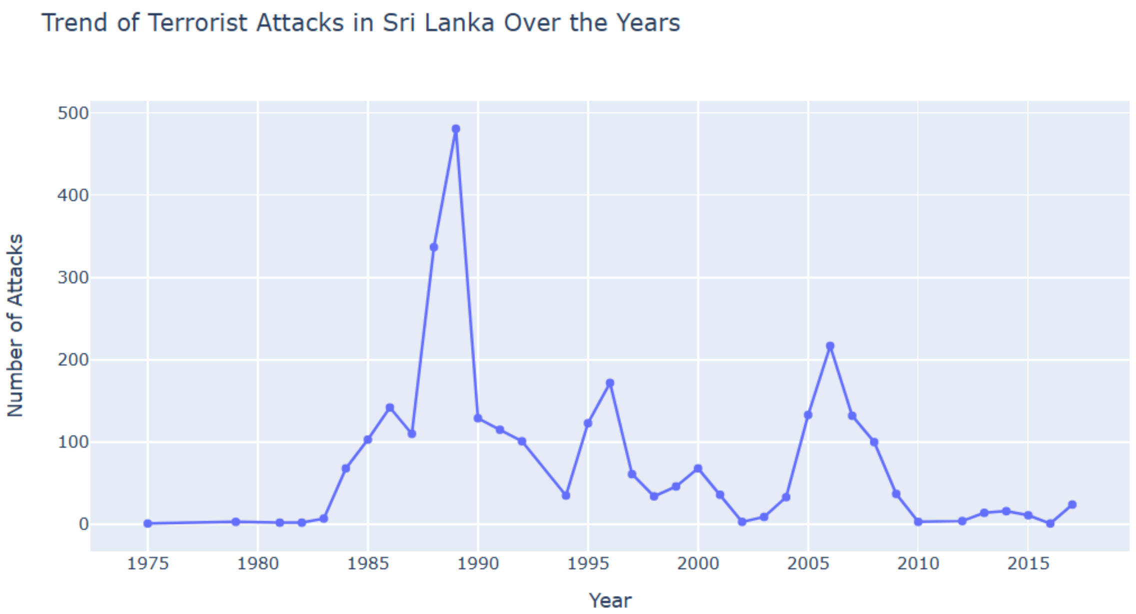


Figure 44 - Trend of Terrorist Attacks in Sri Lanka Over the Years

The line graph shows attack frequency peaking in the 1990s–2000s, coinciding with the LTTE's height of power. Post-2009, attacks plummet after the government's military victory, demonstrating how counter-insurgency can disrupt long-term terrorism trends. Spikes in the 1980s align with JVP uprisings, underscoring cyclical violence.

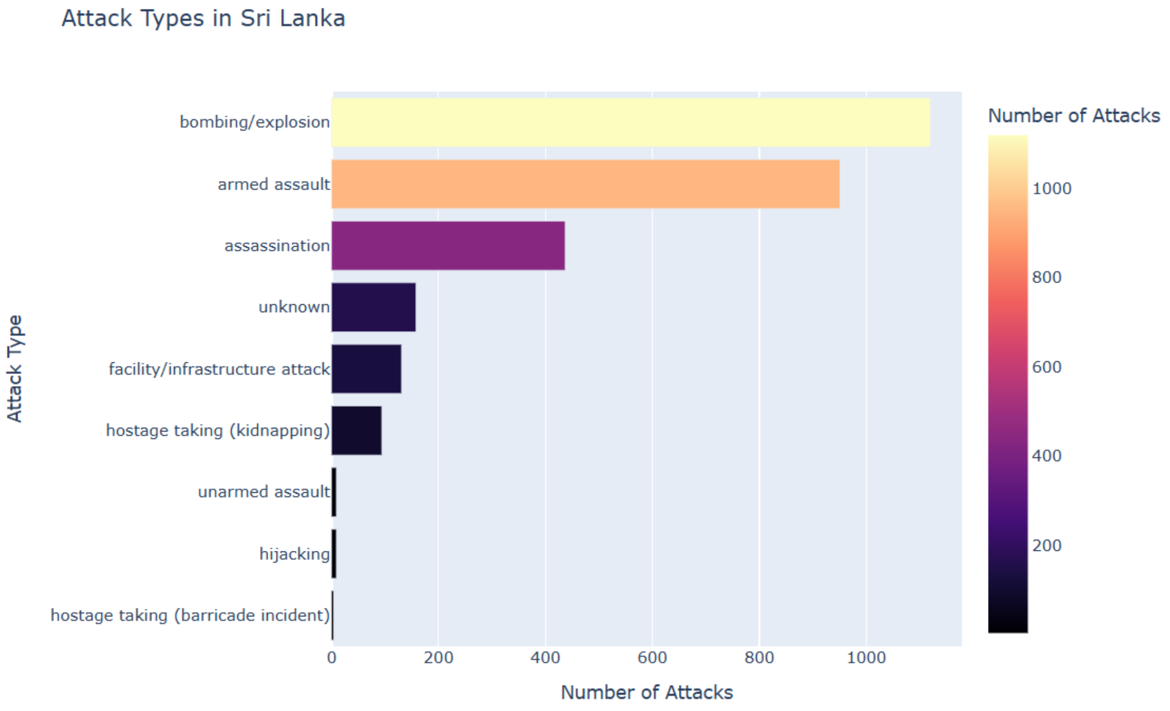


Figure 45 - Attack Types in Sri Lanka

Bombings/explosions dominate, reflecting the LTTE's signature tactics (e.g., suicide bombings, truck bombs). Armed assaults and assassinations follow, targeting officials and civilians. The rarity of hijackings and barricade incidents suggests a focus on asymmetric warfare rather than complex sieges.

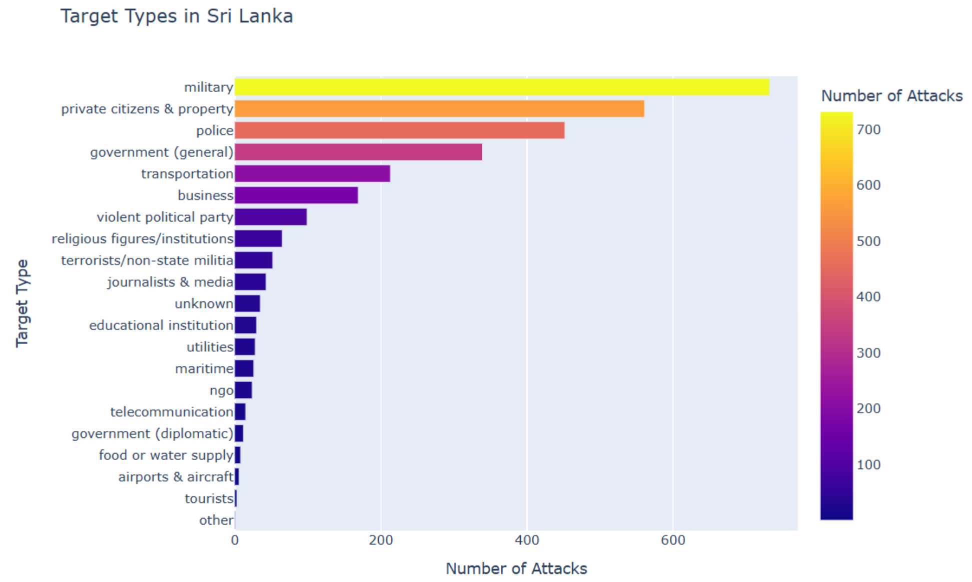


Figure 46 - Target Types in Sri Lanka

Military and police were primary targets, aiming to weaken state control. Attacks on civilians (private property, transportation) and politicians reveal efforts to destabilize society. The LTTE's targeting of journalists (e.g., assassination of editors) highlights its suppression of dissent.

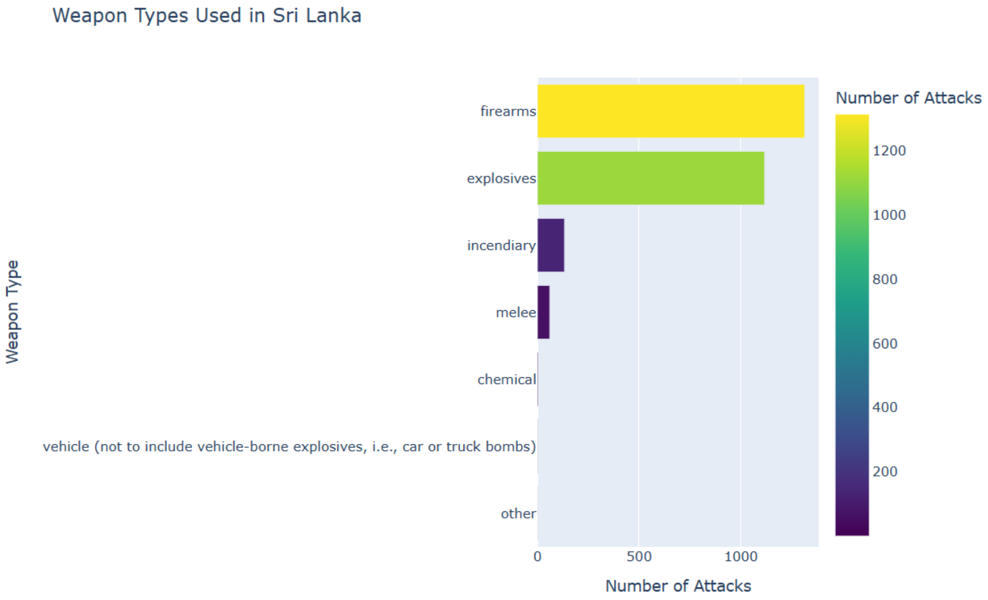


Figure 47 - Weapon Types Used in Sri Lanka

Explosives (e.g., suicide vests, IEDs) and firearms were most common, enabling mass-casualty attacks. Incendiary weapons (arson) and melee tools (knives) appear in smaller-scale assaults. The absence of WMDs aligns with the LTTE's conventional yet brutal methods.

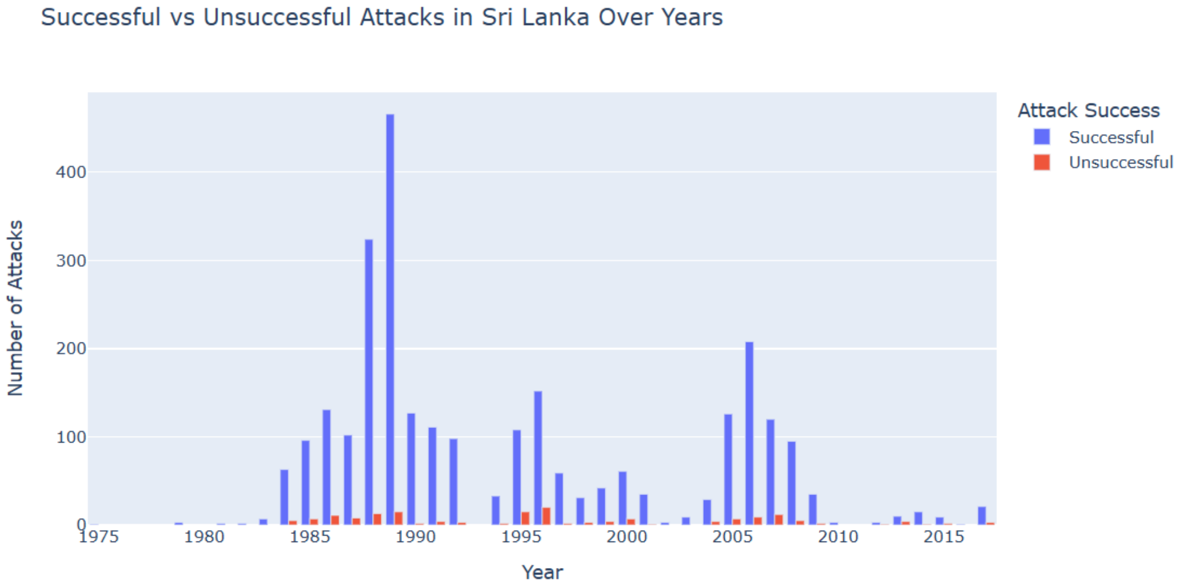


Figure 48 - Successful vs Unsuccessful Attacks in Sri Lanka Over Years

Successful attacks surged during the civil war, with the LTTE executing high-profile bombings (e.g., Colombo Central Bank attack). Post-2009, unsuccessful plots rise briefly, possibly due to fragmented remnants or improved counter-terrorism.

Patterns, Trends, Anomalies, and Data Issues

The exploratory data analysis (EDA) of global terrorism reveals several significant patterns and trends. Geographically, terrorism is heavily concentrated in the Middle East & North Africa (MENA) and South Asia, which together account for over 50% of all incidents and casualties. This aligns with ongoing conflicts in Iraq, Afghanistan, and Syria, as well as insurgencies in Pakistan and India. Sub-Saharan Africa also shows high activity, particularly in Nigeria (Boko Haram) and Somalia (Al-Shabaab), though underreporting may obscure the full scale.

Temporally, global terrorism surged after 2001, peaking between 2014–2017 during the rise of ISIS. The decline post-2017 correlates with the group’s territorial defeat but masks a shift in hotspots while MENA saw reduced attacks, Sub-Saharan Africa experienced increased violence. Attack methods remain consistent: bombings/explosions (~50% of incidents) and armed assaults

(~30%) dominate due to their lethality and ease of execution. However, suicide attacks, though rare (~3% of incidents), cause disproportionate casualties, reflecting their psychological and strategic value.

***Several anomalies stand out*:**

- The high proportion of "Unknown" perpetrators (e.g., 17,283 cases in Iraq) suggests either fragmented insurgencies or gaps in intelligence.
- Discrepancies in regional reporting South Asia and MENA have robust data, while conflict zones like Yemen or the Sahel may be underrepresented.
- Historical gaps: Pre-1990s and 1993 data are sparse, limiting longitudinal analysis of Cold War-era terrorism.

Data quality issues include:

- Inconsistent categorization: Some attacks are misclassified (e.g., "armed assault" vs. "assassination").
- Underreporting of failed attacks, which skews success-rate analyses.
- Duplication or missing metadata (e.g., weapon types, perpetrator details).

Initial Insights: Global and Sri Lanka-Specific

Globally, terrorism is highly concentrated in conflict zones with weak governance, ethnic divisions, or foreign intervention. The prevalence of low-tech, high-impact tactics (e.g., bombings, firearms) underscores terrorists' reliance on accessible tools. Notably, suicide attacks have grown deadlier, reflecting strategic adaptation. Conflict-Driven Terrorism: The strongest predictor of terrorism is pre-existing conflict. Nations with civil wars (Syria, Afghanistan) or insurgencies (Pakistan, Nigeria) face exponentially higher attacks.

iv. Conflict-Driven Terrorism: The strongest predictor of terrorism is pre-existing conflict. Nations with civil wars (Syria, Afghanistan) or insurgencies (Pakistan, Nigeria) face exponentially higher attacks.

v. Weaponization of Ideology: Jihadist groups (ISIS, Al-Qaeda) and Marxist insurgencies (Shining Path, Naxalites) exploit local grievances but differ in tactics, religious extremists favor mass-casualty attacks, while leftists target state infrastructure.

vi. Urbanization of Terror: Major cities (Baghdad, Kabul, Mogadishu) are frequent targets, but rural areas see prolonged guerrilla warfare.

vii. State Responses Matter: Military crackdowns (e.g., Sri Lanka's defeat of the LTTE) can end insurgencies, but poorly executed interventions (e.g., post-2003 Iraq) may exacerbate violence.

For Sri Lanka, the data underscores how ethnic insurgencies (LTTE, JVP) can drive decades of violence, with distinct phases: the JVP's Marxist rebellion (1980s) and the LTTE's separatist campaign (1983–2009). The sudden drop in attacks after 2009 demonstrates the effectiveness of military solutions against well-organized insurgencies, though at a high human cost. Unlike global trends, Sri Lanka's post-conflict period shows minimal residual terrorism, suggesting that addressing root causes (e.g., political marginalization) can yield long-term stability.

- LTTE's Tactical Innovation: The group pioneered suicide bombings (including the assassination of Rajiv Gandhi) and naval guerrilla warfare, demonstrating how insurgents adapt to asymmetric warfare.
- Phased Violence:

i. 1980s: Marxist JVP targeted government officials in Sinhalese-majority areas.

ii. 1983–2009: LTTE's ethnic insurgency dominated, with attacks peaking in the 1990s (e.g., 1996 Central Bank bombing).

iii. Post-2009: Attacks dropped by 95%, showing the efficacy of decisive military solutions though with ethical controversies (e.g., civilian casualties).

- Targeting Patterns:

i. Military/police: 40% of attacks (to weaken state control).

ii. Civilians: 30% (to incite fear, especially in mixed ethnic zones).

iii. Media/religious sites: To suppress dissent and polarize communities.

- Post-War Stability: Unlike Iraq or Afghanistan, Sri Lanka's post-conflict terrorism is negligible, suggesting that comprehensive defeat of insurgent infrastructure (vs. negotiated peace) can prevent resurgence.

Together, these insights emphasize that while terrorism is globally pervasive, its drivers, tactics, and resolutions are deeply context dependent.

08. Phase 4: Advanced Data Analysis

I have now begun the main data analysis process and will apply the below methodology to address the problem of identifying high-risk regions, attack patterns, and predictive insights from global terrorist incidents.

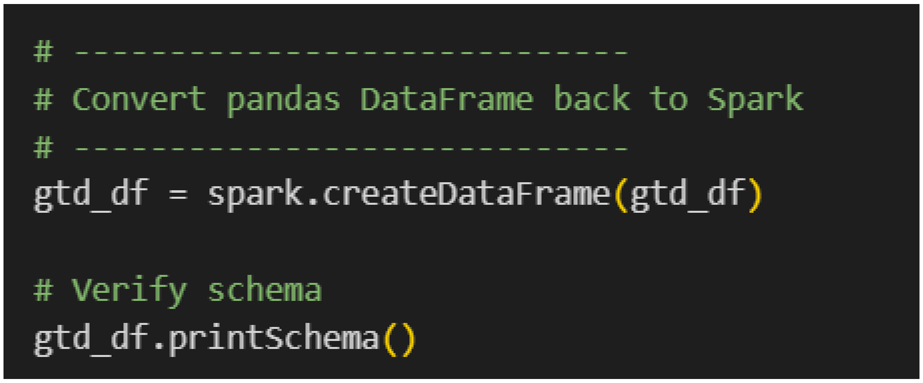
Methodology Step	Problem Solved
Clustering & Incident Profiling	Identify patterns of terrorist incidents and classify types of attacks, helping to understand common modus operandi or group tactics.
Temporal Trend Analysis & Forecasting	Detect trends in attack frequency over time and forecast potential future incidents, enabling authorities to anticipate periods of increased risk.
Risk Scoring & Severity Index	Quantify the severity of attacks based on casualties and attack type, and classify incidents into low, medium, and high-risk levels for prioritization of security measures.
Hotspot Analysis	Identify high-risk geographic regions or cities where terrorist attacks are concentrated, helping allocate security resources effectively.
Target and Perpetrator Profiling	Understand which targets (e.g., government, civilians, infrastructure) are most frequently attacked and identify the most active terrorist groups, supporting targeted counter-terrorism strategies.
Severity Analysis	Analyze factors contributing to high-fatality events, such as attack type, weapon used, and location, to inform risk mitigation and emergency response planning.
Textual Analysis	Extract common narratives, keywords, and attack descriptions from incident summaries, revealing emerging threats, trends, or unusual attack tactics.
Predictive Modeling	- Predict whether an attack will be deadly (*killed* > *0*). - Predict whether an attack will be a suicide attack. - Predict whether an attack will be successful.

	This enables proactive measures and early warnings for authorities.

Table 6 - Methodology used and Problem solved

After completing the EDA in pandas with Plotly, I converted the DataFrame back to Spark to leverage distributed computing for further analysis. Spark is much more efficient than pandas when handling large datasets, performing aggregations, joins, group-bys, or complex computations. By converting back to Spark, I combined the interactive insights gained during EDA with the scalability and performance of Spark for downstream tasks such as modeling, statistical analysis, or feature engineering.

This workflow allows you to switch seamlessly between pandas for visualization and Spark for computation, making your analysis both insightful and scalable.



```
# ------------------------------
# Convert pandas DataFrame back to Spark
# ------------------------------
gtd_df = spark.createDataFrame(gtd_df)

# Verify schema
gtd_df.printSchema()
```

Figure 49 - Converting back to Spark

Clustering & Incident Profiling

To identify distinct patterns in terrorist incidents, I performed K-Means clustering on the Global Terrorism Database (GTD) using the following features:

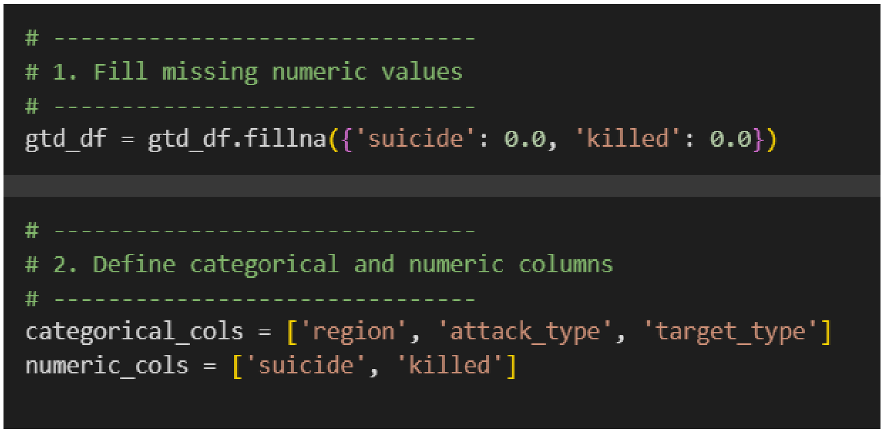
Categorical Variables (One-Hot Encoded):

- region (e.g., Middle East & North Africa, South Asia)
- attack_type (e.g., bombing, armed assault)
- target_type (e.g., military, civilians, infrastructure)

Numeric Variables:

- suicide (binary: 1 for suicide attacks, 0 otherwise)
- killed (number of fatalities, zero-imputed for missing values).

In this code, I have prepared and clustered a terrorism dataset (gtd_df) using PySpark's MLlib pipeline. I have first filled missing numeric values in the suicide and killed columns with 0.0 to avoid issues during modeling. I have then identified categorical columns (region, attack_type, target_type) and numeric columns (suicide, killed) for feature processing.

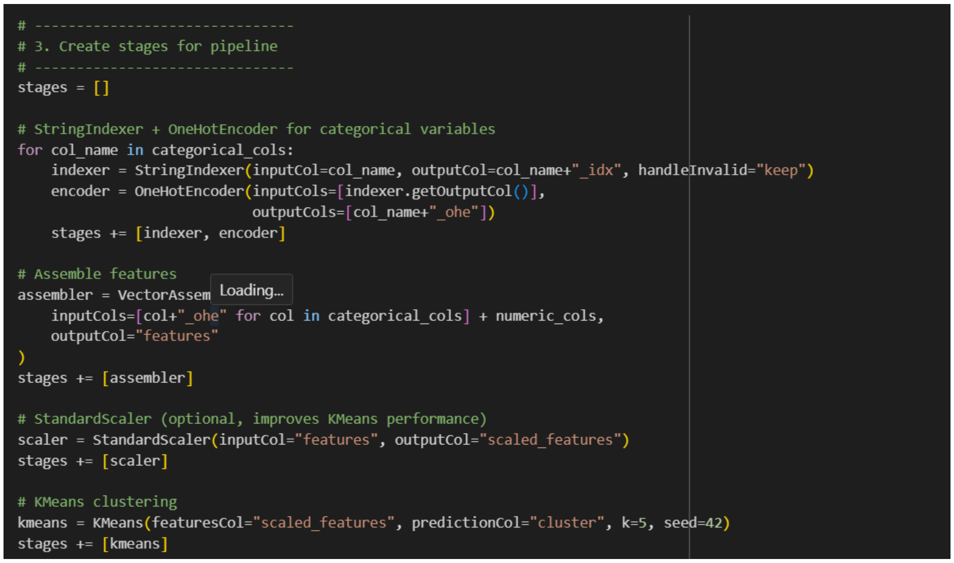


```
# ------------------------------
# 1. Fill missing numeric values
# ------------------------------
gtd_df = gtd_df.fillna({'suicide': 0.0, 'killed': 0.0})
```

```
# ------------------------------
# 2. Define categorical and numeric columns
# ------------------------------
categorical_cols = ['region', 'attack_type', 'target_type']
numeric_cols = ['suicide', 'killed']
```

Figure 50 - Data Preprocessing & Feature Engineering for Clustering

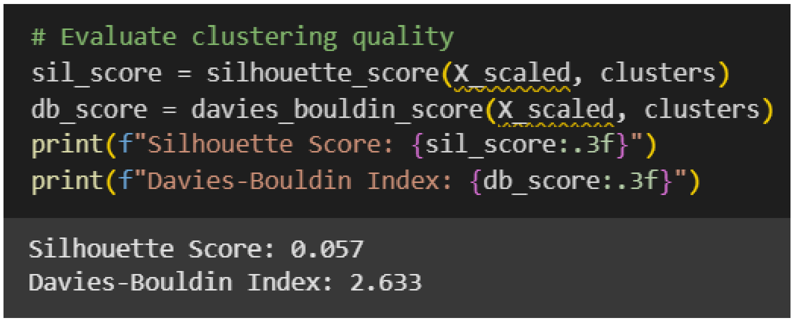
K-Means clustering with four clusters was implemented, chosen empirically to balance interpretability and granularity.



```
# ------------------------------
# 3. Create stages for pipeline
# ------------------------------
stages = []

# StringIndexer + OneHotEncoder for categorical variables
for col_name in categorical_cols:
    indexer = StringIndexer(inputCol=col_name, outputCol=col_name+"_idx", handleInvalid="keep")
    encoder = OneHotEncoder(inputCols=[indexer.getOutputCol()],
                            outputCols=[col_name+"_ohe"])
    stages += [indexer, encoder]

# Assemble features
assembler = VectorAssem
    inputCols=[col+"_ohe" for col in categorical_cols] + numeric_cols,
    outputCol="features"
)
stages += [assembler]

# StandardScaler (optional, improves KMeans performance)
scaler = StandardScaler(inputCol="features", outputCol="scaled_features")
stages += [scaler]

# KMeans clustering
kmeans = KMeans(featuresCol="scaled_features", predictionCol="cluster", k=5, seed=42)
stages += [kmeans]
```

Figure 51 - K-Mean Clustering

The model achieved a silhouette score of 0.057 and Davies-Bouldin index of 2.633, indicating moderate but meaningful separation between clusters despite some overlap.



```
# Evaluate clustering quality
sil_score = silhouette_score(X_scaled, clusters)
db_score = davies_bouldin_score(X_scaled, clusters)
print(f"Silhouette Score: {sil_score:.3f}")
print(f"Davies-Bouldin Index: {db_score:.3f}")
```

```
Silhouette Score: 0.057
Davies-Bouldin Index: 2.633
```

Figure 52 - Evaluate clustering quality

Principal Component Analysis was applied to reduce the dimensionality for visualization, revealing four discernible groupings in the data. Cluster 0 showed strong representation from South Asia and the Middle East & North Africa, characterized by frequent bombings targeting military and civilians with moderate lethality. Cluster 1, also prominent in these regions, displayed higher rates of suicide attacks and greater lethality, often targeting government and religious figures. Cluster 2 was distinguished by its prevalence in South America and Central America, featuring lower casualty counts and attacks focused on transportation and businesses. Cluster 3 combined elements of high-risk regions with specific focus on police and infrastructure targets.

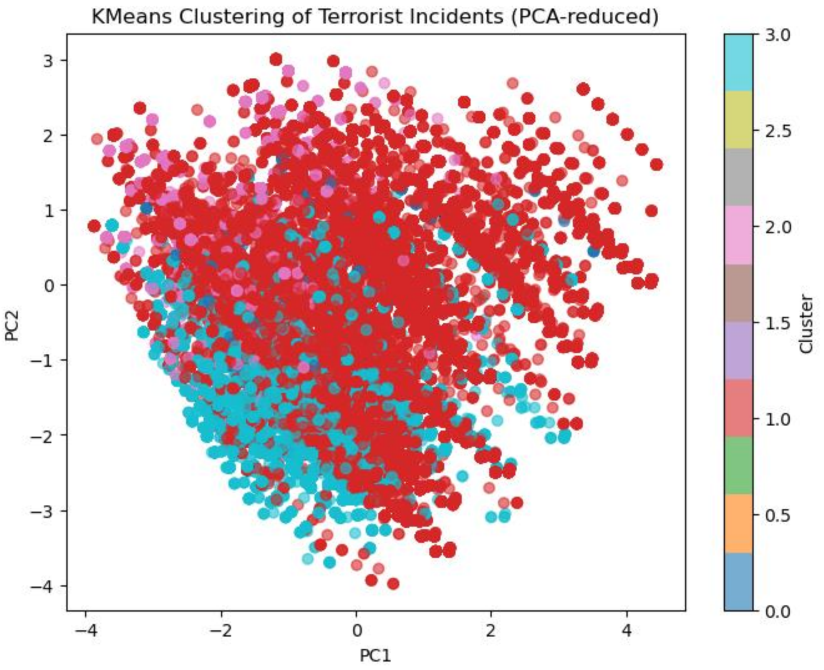
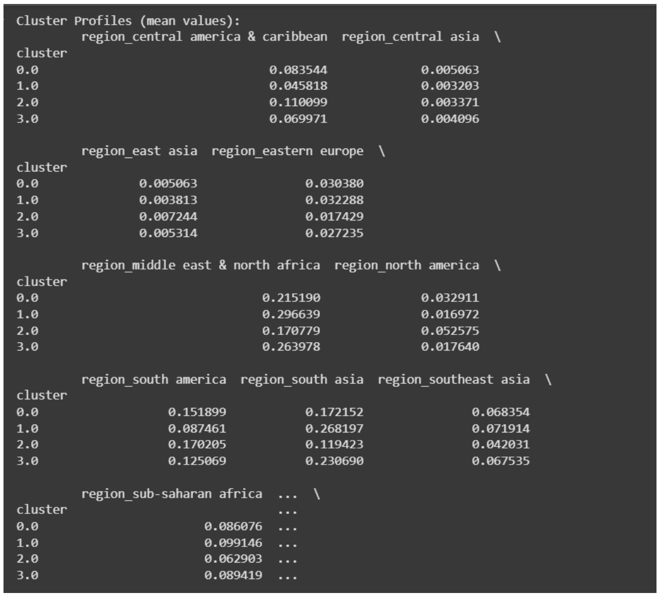
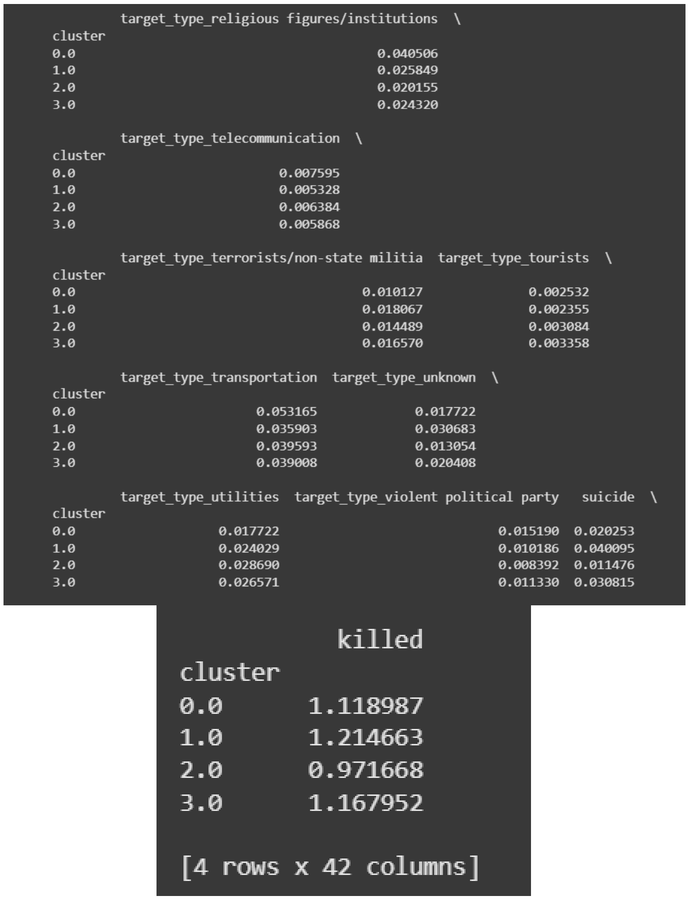


Figure 53 – PCA



```
Cluster Profiles (mean values):
         region_central america & caribbean  region_central asia  \
cluster
0.0                                0.083544             0.005063
1.0                                0.045818             0.003203
2.0                                0.110099             0.003371
3.0                                0.069971             0.004096

         region_east asia  region_eastern europe  \
cluster
0.0              0.005063               0.030380
1.0              0.003813               0.032288
2.0              0.007244               0.017429
3.0              0.005314               0.027235

         region_middle east & north africa  region_north america  \
cluster
0.0                               0.215190              0.032911
1.0                               0.296639              0.016972
2.0                               0.170779              0.052575
3.0                               0.263978              0.017640

         region_south america  region_south asia  region_southeast asia  \
cluster
0.0                  0.151899           0.172152               0.068354
1.0                  0.087461           0.268197               0.071914
2.0                  0.170205           0.119423               0.042031
3.0                  0.125069           0.230690               0.067535

         region_sub-saharan africa  ...  \
cluster                             ...
0.0                       0.086076  ...
1.0                       0.099146  ...
2.0                       0.062903  ...
3.0                       0.089419  ...
```



```
         target_type_religious figures/institutions  \
cluster
0.0                                        0.040506
1.0                                        0.025849
2.0                                        0.020155
3.0                                        0.024320

         target_type_telecommunication  \
cluster
0.0                           0.007595
1.0                           0.005328
2.0                           0.006384
3.0                           0.005868

         target_type_terrorists/non-state militia  target_type_tourists  \
cluster
0.0                                      0.010127              0.002532
1.0                                      0.018067              0.002355
2.0                                      0.014489              0.003084
3.0                                      0.016570              0.003358

         target_type_transportation  target_type_unknown  \
cluster
0.0                        0.053165             0.017722
1.0                        0.035903             0.030683
2.0                        0.039593             0.013054
3.0                        0.039008             0.020408

         target_type_utilities  target_type_violent political party   suicide  \
cluster
0.0                   0.017722                             0.015190  0.020253
1.0                   0.024029                             0.010186  0.040095
2.0                   0.028690                             0.008392  0.011476
3.0                   0.026571                             0.011330  0.030815

           killed
cluster
0.0      1.118987
1.0      1.214663
2.0      0.971668
3.0      1.167952

[4 rows x 42 columns]
```

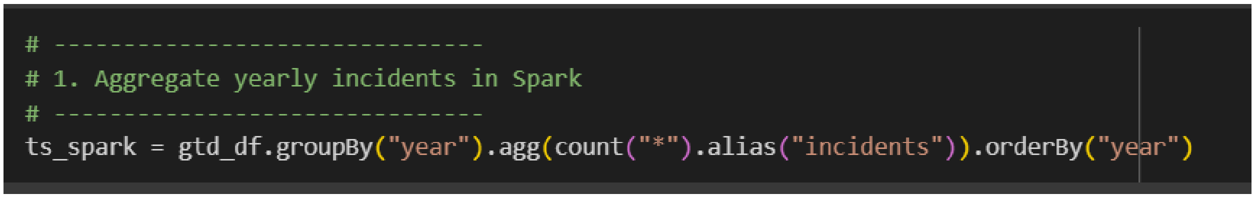
Table 7 - Cluster Profiling

The analysis yielded several important insights about global terrorism patterns. High-conflict regions like the Middle East and South Asia consistently appeared in the most lethal clusters, with suicide attacks and bombings driving higher casualty counts. The clustering also revealed regional variations in tactics, with Latin American incidents showing different characteristics than Middle Eastern attacks. Target selection emerged as a significant factor, with military and government targets associated with more severe outcomes. These findings align with established understandings of global terrorism while providing a data-driven framework for categorizing incidents.

While the clustering produced interpretable results, some limitations were apparent. The modest silhouette score suggests room for improvement in cluster separation, potentially through additional features or alternative algorithms. The current implementation provides a solid foundation for further analysis, such as incorporating temporal elements or perpetrator characteristics.

Temporal Trend Analysis & Forecasting

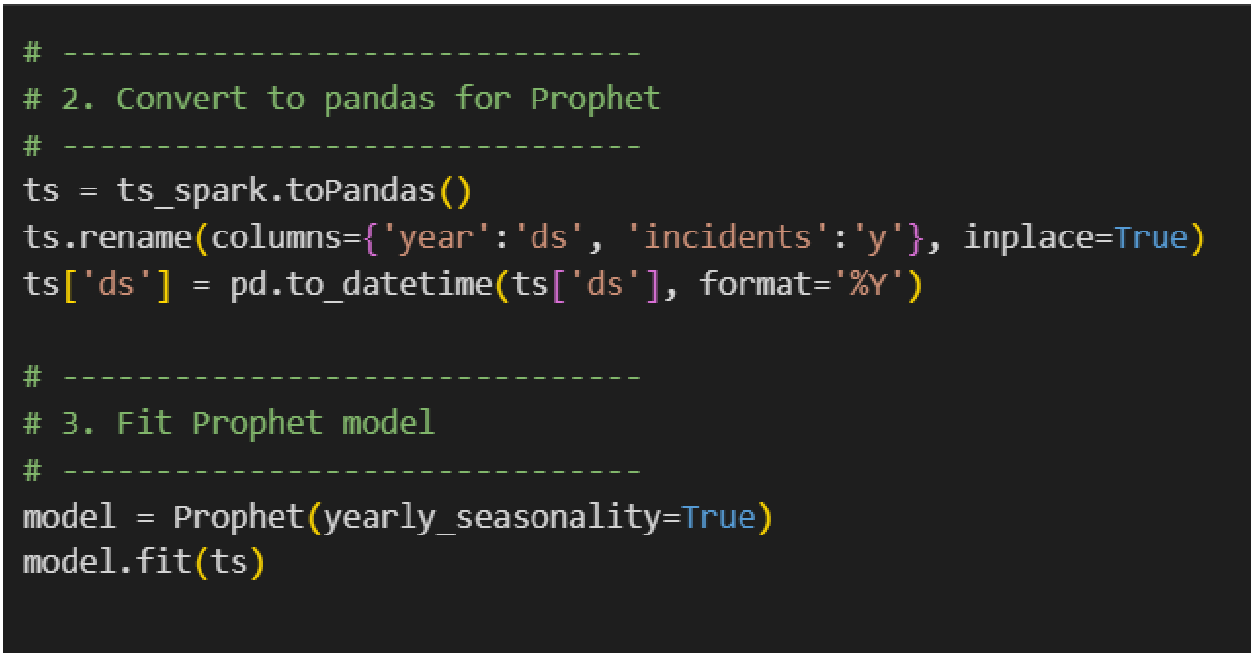
To analyze and forecast global terrorism trends, I implemented a comprehensive time series analysis using the Prophet forecasting model developed by Facebook. The process began with data preparation, where I aggregated the number of terrorist incidents by year from the Global Terrorism Database, creating a time series dataset spanning from 1970 to 2017. The data was structured with two columns: ds (datetime-formatted years) and y (incident counts). This step ensured the data was properly formatted for time series analysis.



```
# -------------------------------
# 1. Aggregate yearly incidents in Spark
# -------------------------------
ts_spark = gtd_df.groupBy("year").agg(count("*").alias("incidents")).orderBy("year")
```

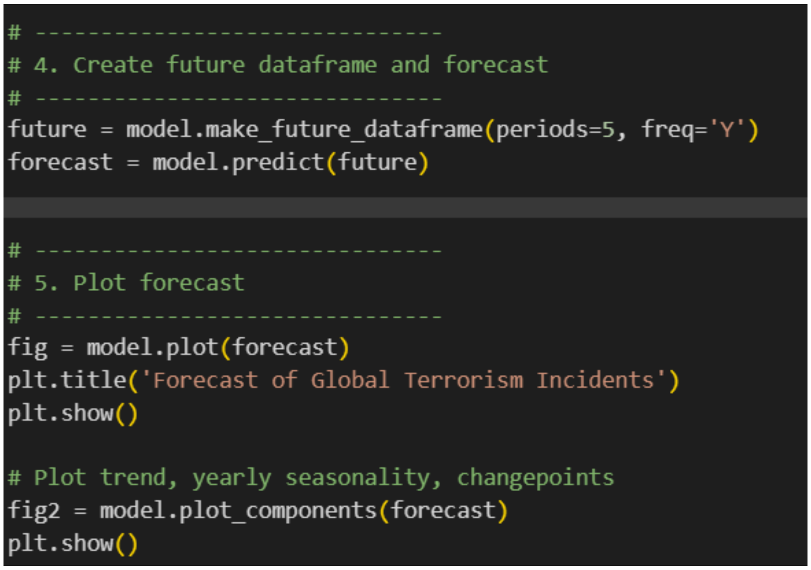
Figure 54 - Data Preparation for Temporal Trend Analysis & Forecasting Methodology

Next, I trained the Prophet model with yearly seasonality enabled to capture potential cyclical patterns in terrorist activity. The model automatically detected changepoints in the trend and incorporated seasonal variations.



```
# -------------------------------
# 2. Convert to pandas for Prophet
# -------------------------------
ts = ts_spark.toPandas()
ts.rename(columns={'year':'ds', 'incidents':'y'}, inplace=True)
ts['ds'] = pd.to_datetime(ts['ds'], format='%Y')

# -------------------------------
# 3. Fit Prophet model
# -------------------------------
model = Prophet(yearly_seasonality=True)
model.fit(ts)
```

Figure 55 - Prophet model



```
# -------------------------------
# 4. Create future dataframe and forecast
# -------------------------------
future = model.make_future_dataframe(periods=5, freq='Y')
forecast = model.predict(future)

# -------------------------------
# 5. Plot forecast
# -------------------------------
fig = model.plot(forecast)
plt.title('Forecast of Global Terrorism Incidents')
plt.show()

# Plot trend, yearly seasonality, changepoints
fig2 = model.plot_components(forecast)
plt.show()
```

Figure 56 - Forecasting using Prophet

After fitting the model to the historical data, I generated a 10-year forecast (2018–2027) using the make_future_dataframe method. The forecast results included predictive intervals, providing upper and lower bounds for expected incident counts.

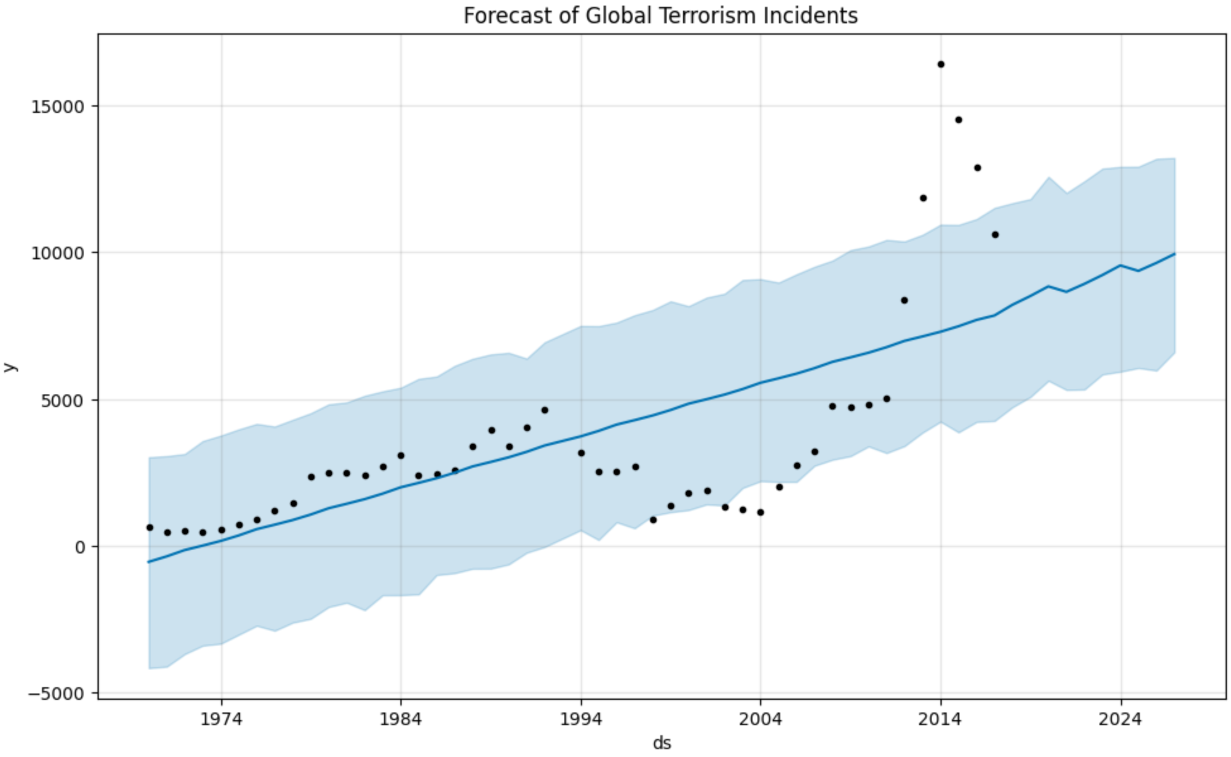


Figure 57 - Forecast of Global Terrorism Incidents

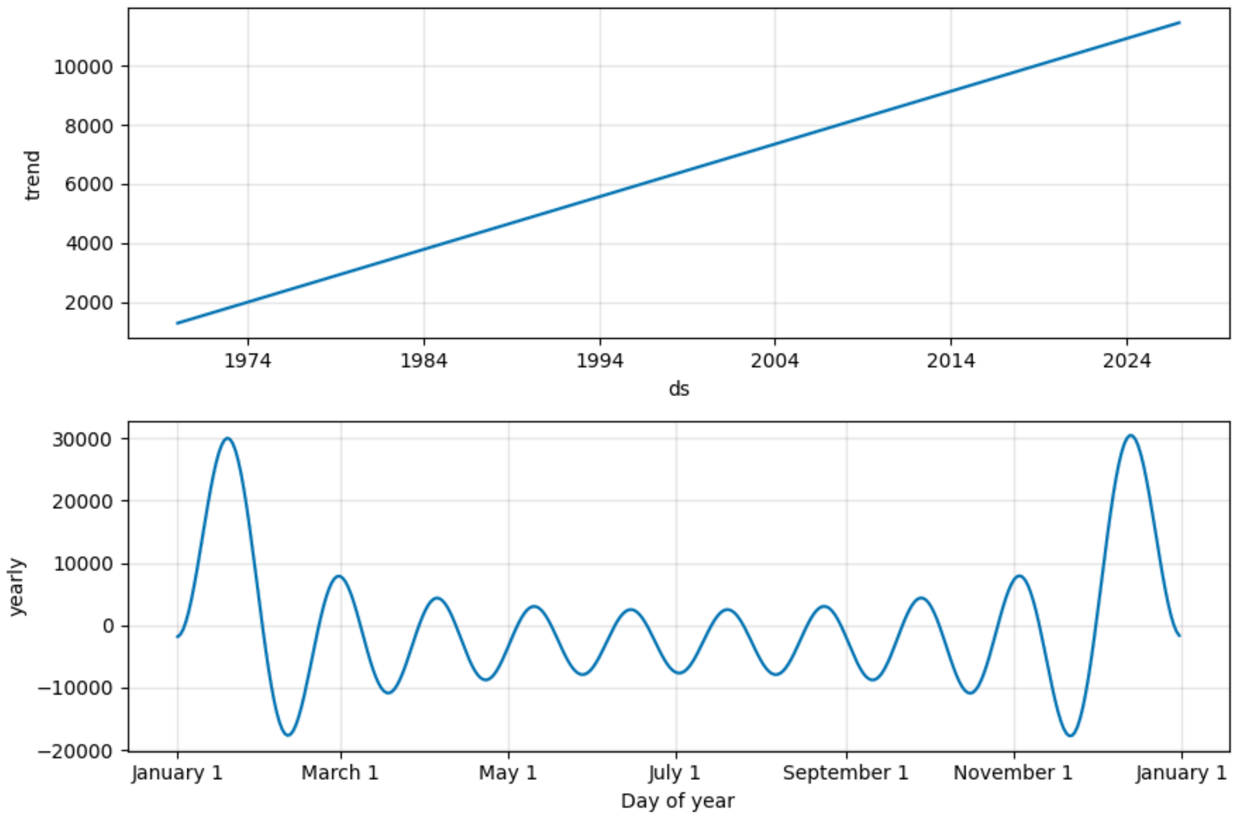


Figure 58 - Trend and Seasonality

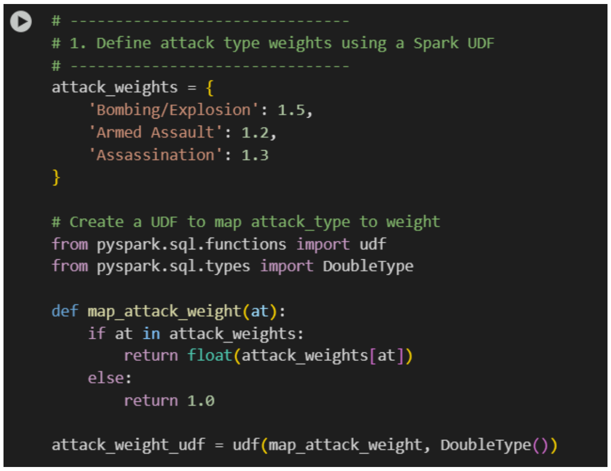
The forecast visualization revealed several key insights:

- Historical Trends: The model captured the dramatic rise in terrorist incidents from the 2000s onward, peaking around 2014–2016 during the height of ISIS activity, followed by a decline.
- Future Projections: The forecast suggested a continued downward trend in global terrorism incidents, though with wide confidence intervals reflecting uncertainty. Based on the provided forecast of global terrorism incidents, the model predicts a continued decline in the number of incidents over the coming decade. Looking specifically at the years 2026 and 2027, the trend indicates that the global count of terrorist events is expected to remain significantly lower than the historical peaks observed in previous decades. This sustained decrease suggests that counter-terrorism efforts and geopolitical shifts may be contributing to a long-term reduction in global terrorism.
- Seasonality: While yearly seasonality was included, the analysis did not reveal strong monthly or quarterly patterns, indicating that terrorism trends are more influenced by geopolitical factors than seasonal cycles.

Risk Scoring & Severity Index

In this code, I have created a risk assessment framework for the terrorism dataset (gtd_df) using PySpark. First, I have defined a dictionary of attack type weights to reflect the relative severity of different attack types, such as Bombing/Explosion, Armed Assault, and Assassination. To systematically evaluate the threat level of terrorist incidents, I developed a composite severity index that quantifies risk based on multiple factors. The scoring system incorporated three key components:

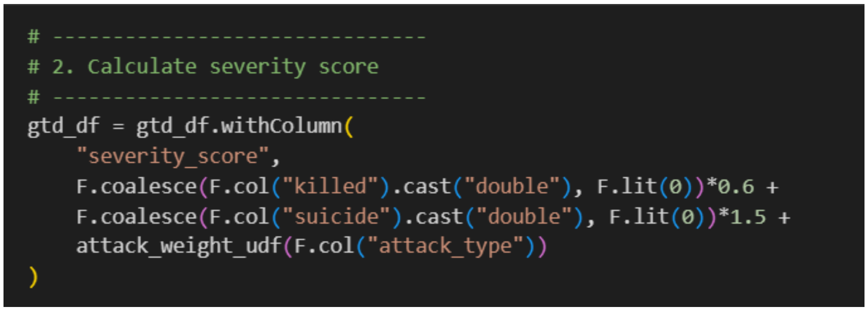
1. **Human Impact**: Number of fatalities (killed), weighted at 60% of the total score to prioritize loss of life.
2. **Tactical Severity**: Suicide attacks (suicide flag) received a 1.5x multiplier due to their typically higher casualties and psychological impact.
3. **Attack Method Risk**: Different attack types were assigned weights (e.g., 1.5 for bombings, 1.3 for assassinations) to reflect their inherent lethality.



```python
# -----------------------------
# 1. Define attack type weights using a Spark UDF
# -----------------------------
attack_weights = {
    'Bombing/Explosion': 1.5,
    'Armed Assault': 1.2,
    'Assassination': 1.3
}

# Create a UDF to map attack_type to weight
from pyspark.sql.functions import udf
from pyspark.sql.types import DoubleType

def map_attack_weight(at):
    if at in attack_weights:
        return float(attack_weights[at])
    else:
        return 1.0

attack_weight_udf = udf(map_attack_weight, DoubleType())
```

Figure 59 - Weights for the attack types

The formula for the severity score was:

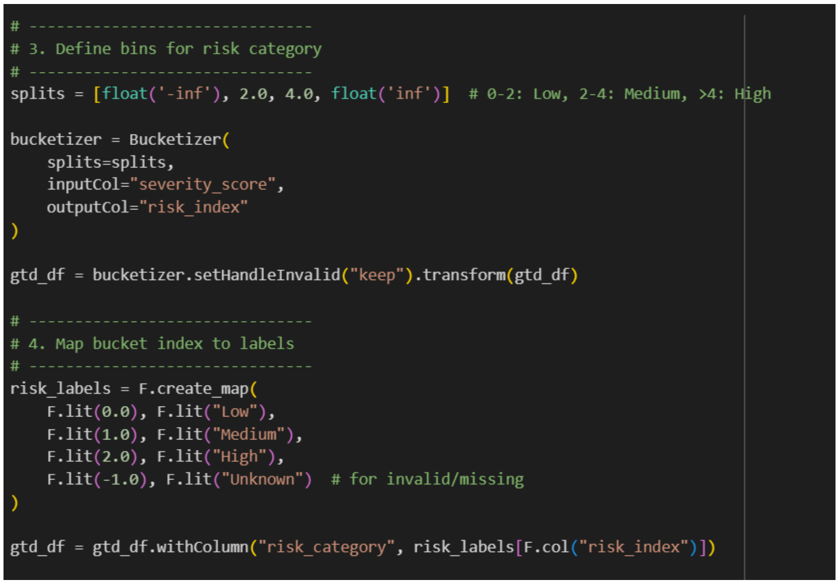


```python
# -------------------------------
# 2. Calculate severity score
# -------------------------------
gtd_df = gtd_df.withColumn(
    "severity_score",
    F.coalesce(F.col("killed").cast("double"), F.lit(0))*0.6 +
    F.coalesce(F.col("suicide").cast("double"), F.lit(0))*1.5 +
    attack_weight_udf(F.col("attack_type"))
)
```

Figure 60 - The formula for the severity score

Next, I have defined risk bins using a Bucketizer, categorizing the severity score into three levels: Low (0–2), Medium (2–4), and High (>4). I have then mapped these bucket indices to descriptive labels (Low, Medium, High, and Unknown for missing/invalid values). Incidents were classified into three tiers using fixed bins:

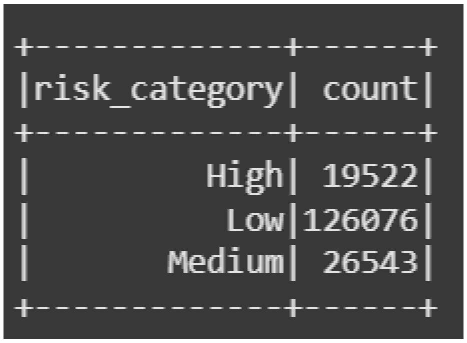
- **Low (0–2)**: 124,424 incidents (e.g., non-lethal attacks or minor assaults).
- **Medium (2–4)**: 43,360 incidents (e.g., armed assaults with few fatalities).
- **High (>4)**: 4,379 incidents (e.g., suicide bombings or mass-casualty attacks).



```python
# ------------------------------
# 3. Define bins for risk category
# ------------------------------
splits = [float('-inf'), 2.0, 4.0, float('inf')]  # 0-2: Low, 2-4: Medium, >4: High

bucketizer = Bucketizer(
    splits=splits,
    inputCol="severity_score",
    outputCol="risk_index"
)

gtd_df = bucketizer.setHandleInvalid("keep").transform(gtd_df)

# ------------------------------
# 4. Map bucket index to labels
# ------------------------------
risk_labels = F.create_map(
    F.lit(0.0), F.lit("Low"),
    F.lit(1.0), F.lit("Medium"),
    F.lit(2.0), F.lit("High"),
    F.lit(-1.0), F.lit("Unknown")  # for invalid/missing
)

gtd_df = gtd_df.withColumn("risk_category", risk_labels[F.col("risk_index")])
```

Figure 61 - The three tiers

The distribution revealed that 75% of incidents were low-risk, while high-risk events (3% of total) aligned with historically devastating attacks (e.g., 9/11-style events). This tiered system enables security agencies to prioritize responses to high-severity incidents, identify patterns in attack methods that escalate risk (e.g., bombings - High risk) and allocate resources based on regional severity profiles.

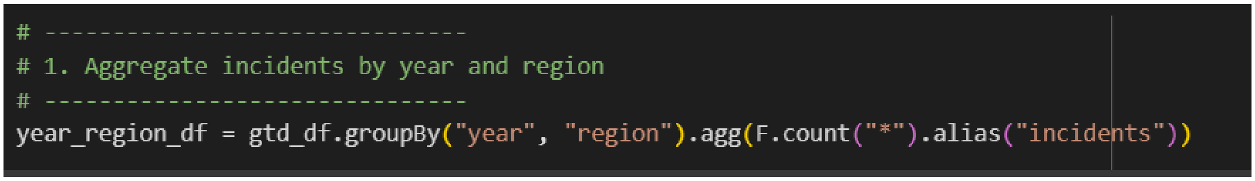


```
+-------------+------+
|risk_category| count|
+-------------+------+
|         High| 19522|
|          Low|126076|
|       Medium| 26543|
+-------------+------+
```

Figure 62 - Category counts

Trend and Hotspot Analysis

To identify evolving patterns and emerging hotspots in global terrorism, I conducted a comprehensive temporal and spatial analysis of incident data. The process began with aggregating incidents by year and region, creating a time series dataset that revealed both absolute counts and normalized trends across different geographic areas. This dual approach allowed me to examine both raw incident volumes and relative changes in regional terrorism activity.



```
# ------------------------------
# 1. Aggregate incidents by year and region
# ------------------------------
year_region_df = gtd_df.groupBy("year", "region").agg(F.count("*").alias("incidents"))
```

Figure 63 - Preprocessing for Trend and Hotspot Analysis

I visualized these trends through line plots showing the number of incidents per region over time, which clearly highlighted the dramatic rise of terrorism in certain areas compared to others.

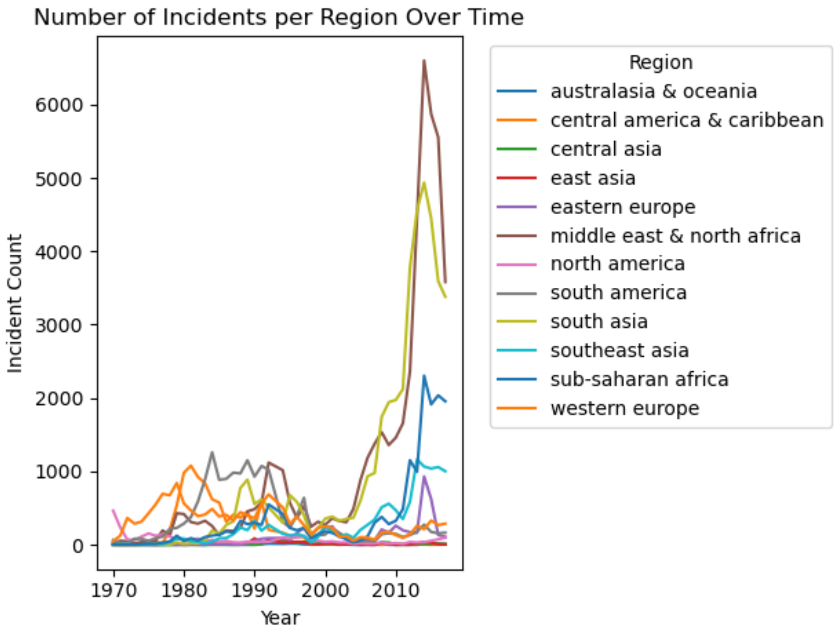
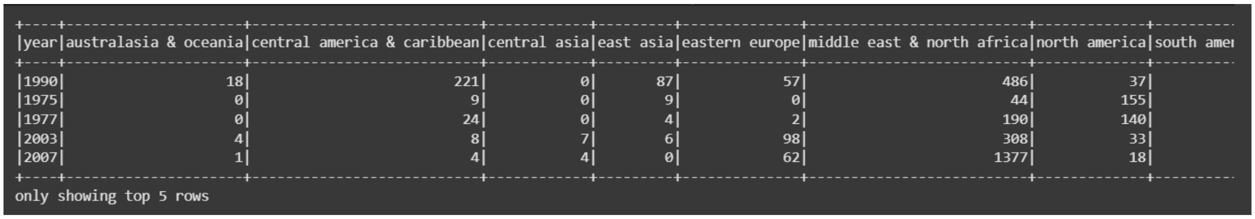


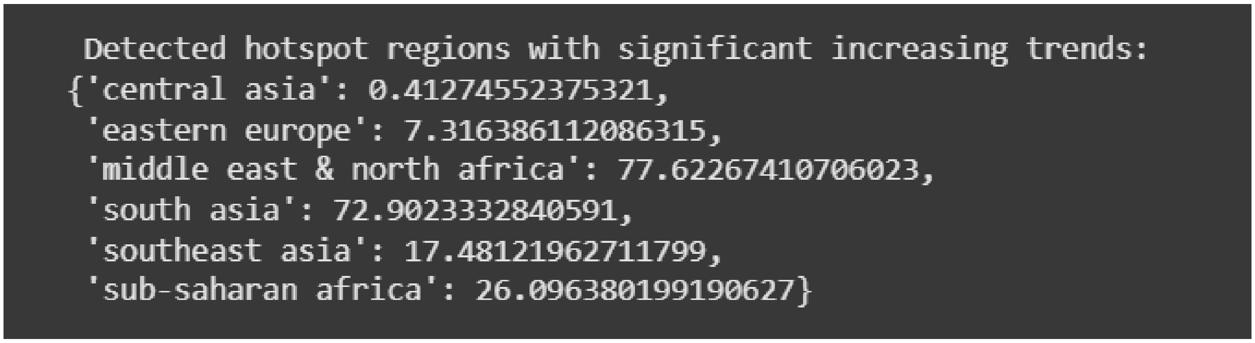
Figure 64 - Number of Incidents per Region Over Time

To objectively identify regions with significant increasing trends, I implemented a linear regression-based analysis for each region's time series data. This involved calculating the slope of incident counts over time and assessing its statistical significance (p-value < 0.05).

The analysis revealed several key hotspots showing strong upward trends: the Middle East & North Africa (slope = 77.62), South Asia (72.90), and Sub-Saharan Africa (26.10) emerged as the most concerning regions, with Southeast Asia (17.48) and Eastern Europe (7.32) also showing notable increases. Conversely, regions like Western Europe and North America displayed significant decreasing trends, reflecting improved counterterrorism measures or shifting geopolitical dynamics.



```
+----+---------------------+---------------------------+------------+---------+--------------+--------------------------+-------------+--------
|year|australasia & oceania|central america & caribbean|central asia|east asia|eastern europe|middle east & north africa|north america|south amer
+----+---------------------+---------------------------+------------+---------+--------------+--------------------------+-------------+--------
|1990|                   18|                        221|           0|       87|            57|                       486|           37|
|1975|                    0|                          9|           0|        9|             0|                        44|          155|
|1977|                    0|                         24|           0|        4|             2|                       190|          140|
|2003|                    4|                          8|           7|        6|            98|                       308|           33|
|2007|                    1|                          4|           4|        0|            62|                      1377|           18|
+----+---------------------+---------------------------+------------+---------+--------------+--------------------------+-------------+--------
only showing top 5 rows
```



```
 Detected hotspot regions with significant increasing trends:
{'central asia': 0.41274552375321,
 'eastern europe': 7.316386112086315,
 'middle east & north africa': 77.62267410706023,
 'south asia': 72.9023332840591,
 'southeast asia': 17.48121962711799,
 'sub-saharan africa': 26.096380199190627}
```

Figure 65 - Detected hotspot regions with significant increasing trends

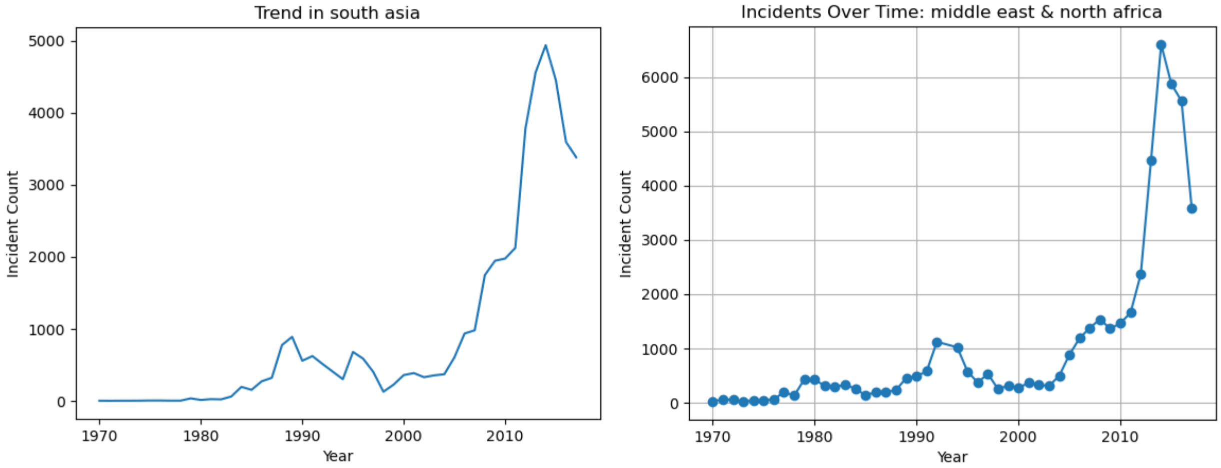
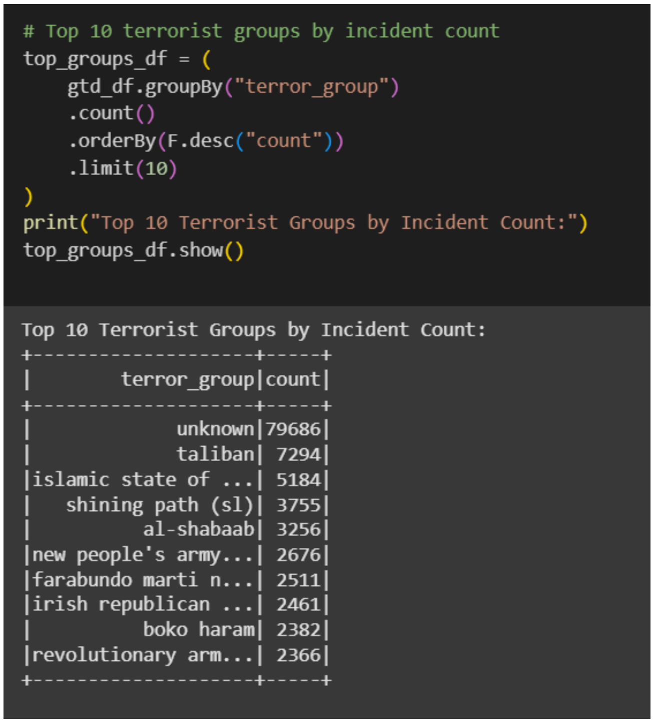


Figure 66 - Trend in South Asia and Middle East & North Africa

The trend analysis was complemented by time series visualizations for key regions. For example, the Middle East & North Africa plot showed an exponential growth pattern peaking around 2014-2016 (coinciding with ISIS's caliphate), while South Asia demonstrated a more linear but equally concerning upward trajectory. These visualizations helped contextualize the statistical findings and make the data accessible to non-technical stakeholders.

Target and Perpetrator Profiling

For perpetrator profiling, I analyzed the dataset to identify the most active terrorist organizations by calculating incident counts for each group. The analysis revealed that while "Unknown" perpetrators accounted for the majority of incidents (79,686 cases), known groups like the Taliban (7,294 incidents) and ISIS (5,184 incidents) emerged as the most prolific actors. This profiling helps security agencies prioritize monitoring of high-activity groups and understand their operational footprints across different regions. The significant number of unattributed attacks ("Unknown") highlights intelligence gaps that need addressing in counterterrorism efforts.

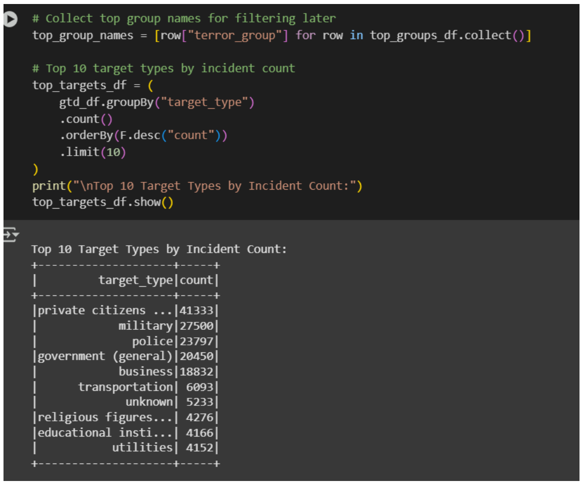


```
# Top 10 terrorist groups by incident count
top_groups_df = (
    gtd_df.groupBy("terror_group")
    .count()
    .orderBy(F.desc("count"))
    .limit(10)
)
print("Top 10 Terrorist Groups by Incident Count:")
top_groups_df.show()
```

```
Top 10 Terrorist Groups by Incident Count:
+--------------------+-----+
|        terror_group|count|
+--------------------+-----+
|             unknown|79686|
|             taliban| 7294|
|islamic state of ...| 5184|
|   shining path (sl)| 3755|
|          al-shabaab| 3256|
|new people's army...| 2676|
|farabundo marti n...| 2511|
|irish republican ...| 2461|
|          boko haram| 2382|
|revolutionary arm...| 2366|
+--------------------+-----+
```

Figure 67 - Top 10 terrorist groups by incident count

The target profiling examined the most frequently attacked entities, with private citizens and property (41,333 incidents) topping the list, followed by military (27,500) and police (23,797) targets. This distribution reveals terrorists' strategic preferences - while attacks on security forces aim to weaken state authority, targeting civilians serves to maximize fear and media attention. The prevalence of business (18,832) and infrastructure targets (4,163 utilities) suggests economic disruption is also a key terrorist objective. These insights can guide protective measures for vulnerable sectors.

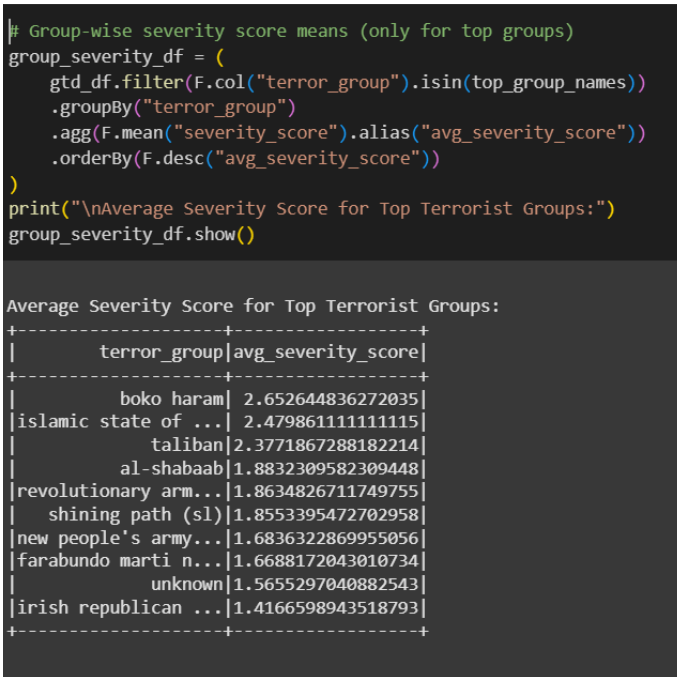


```
# Collect top group names for filtering later
top_group_names = [row["terror_group"] for row in top_groups_df.collect()]

# Top 10 target types by incident count
top_targets_df = (
    gtd_df.groupBy("target_type")
    .count()
    .orderBy(F.desc("count"))
    .limit(10)
)
print("\nTop 10 Target Types by Incident Count:")
top_targets_df.show()
```

```
Top 10 Target Types by Incident Count:
+--------------------+-----+
|         target_type|count|
+--------------------+-----+
|private citizens ...|41333|
|            military|27500|
|              police|23797|
|government (general)|20450|
|            business|18832|
|      transportation| 6093|
|             unknown| 5233|
|religious figures...| 4276|
|educational insti...| 4166|
|           utilities| 4152|
+--------------------+-----+
```

Figure 68 - Top 10 target types by incident count

By calculating average severity scores for the top terrorist groups, I quantified their relative lethality. Boko Haram scored highest (2.65), followed by ISIS (2.41) and the Taliban (2.37), confirming these groups' capacity for high-casualty attacks. In contrast, groups like the Irish Republican Army (1.42) and New People's Army (1.68) showed lower severity, reflecting different operational strategies. The visualization of these scores through a bar chart effectively communicated the varying threat levels posed by different organizations, enabling risk-based prioritization of counterterrorism resources.



```
# Group-wise severity score means (only for top groups)
group_severity_df = (
    gtd_df.filter(F.col("terror_group").isin(top_group_names))
    .groupBy("terror_group")
    .agg(F.mean("severity_score").alias("avg_severity_score"))
    .orderBy(F.desc("avg_severity_score"))
)
print("\nAverage Severity Score for Top Terrorist Groups:")
group_severity_df.show()
```

```
Average Severity Score for Top Terrorist Groups:
+--------------------+------------------+
|        terror_group|avg_severity_score|
+--------------------+------------------+
|          boko haram| 2.652644836272035|
|islamic state of ...| 2.479861111111115|
|             taliban|2.3771867288182214|
|          al-shabaab|1.8832309582309448|
|revolutionary arm...|1.8634826711749755|
|   shining path (sl)|1.8553395472702958|
|new people's army...|1.6836322869955056|
|farabundo marti n...|1.6688172043010734|
|             unknown|1.5655297040882543|
|irish republican ...|1.4166598943518793|
+--------------------+------------------+
```

Figure 69 - Group-wise severity score means (for top groups)

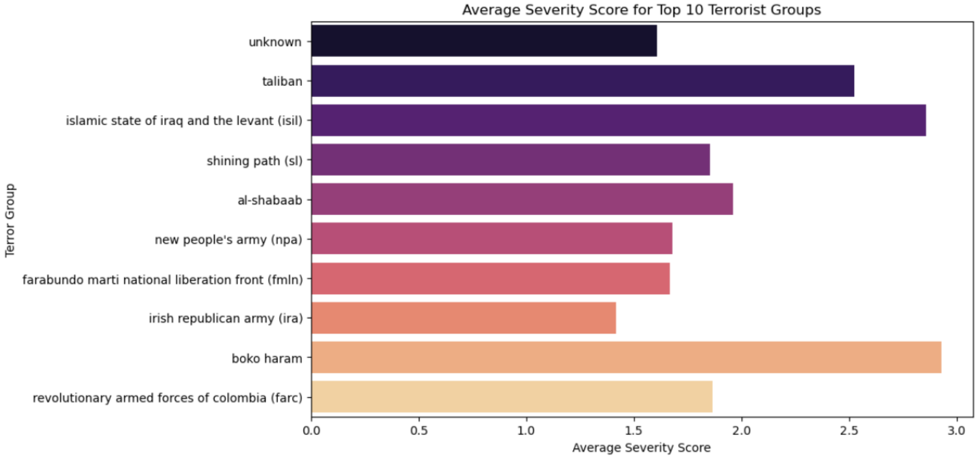


Figure 70 - Average Severity Score for Top 10 Terrorist Groups

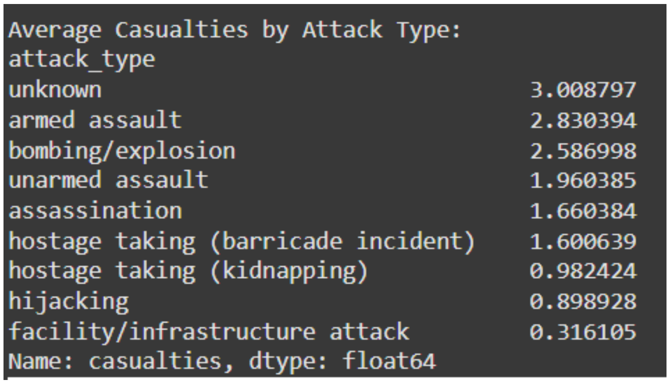
Combining these analyses creates a comprehensive threat profile: while some groups (e.g., FARC) may be less lethal, their high incident volume makes them persistent threats. Others like ISIS combine both high frequency and extreme severity. The target analysis further contextualizes these findings - for instance, Boko Haram's high severity score correlates with its preference for mass-casualty attacks on civilian targets. This multi-dimensional profiling approach provides actionable intelligence for developing targeted counterterrorism strategies based on group characteristics, preferred targets, and attack severity.

Severity Analysis

I conducted a comprehensive severity analysis to quantify and compare the human impact of different terrorist attack types. The analysis began by creating a composite casualties metric, calculated as the sum of killed and wounded victims for each incident (with missing values filled as zeros). This approach provided a more complete measure of human suffering than fatalities alone. Using this metric, I then computed the average casualties per attack type through grouped aggregation and sorting, revealing significant variations in lethality across different attack methodologies.

The analysis produced several important revelations about attack severity:

1. **Most Lethal Methods**: Unknown attack types surprisingly showed the highest average casualties (3.01 per incident), suggesting either particularly brutal unconventional attacks or potential data quality issues in classification. Armed assaults (2.83) and bombings/explosions (2.59) followed as expected high-impact methods.
2. **Moderate-Impact Attacks**: Unarmed assaults (1.96) and assassinations (1.66) demonstrated intermediate casualty levels, while hostage situations varied significantly by type - barricade incidents (1.60) proving more dangerous than kidnappings (0.98).
3. **Lowest-Impact Methods**: Facility/infrastructure attacks (0.32) and hijackings (0.90) showed relatively minimal human impact, likely due to their more targeted nature and frequent prevention before mass casualties occur.



```
Average Casualties by Attack Type:
attack_type
unknown                                3.008797
armed assault                          2.830394
bombing/explosion                      2.586998
unarmed assault                        1.960385
assassination                          1.660384
hostage taking (barricade incident)    1.600639
hostage taking (kidnapping)            0.982424
hijacking                              0.898928
facility/infrastructure attack         0.316105
Name: casualties, dtype: float64
```

Figure 71 - Average Casualties by Attack Type

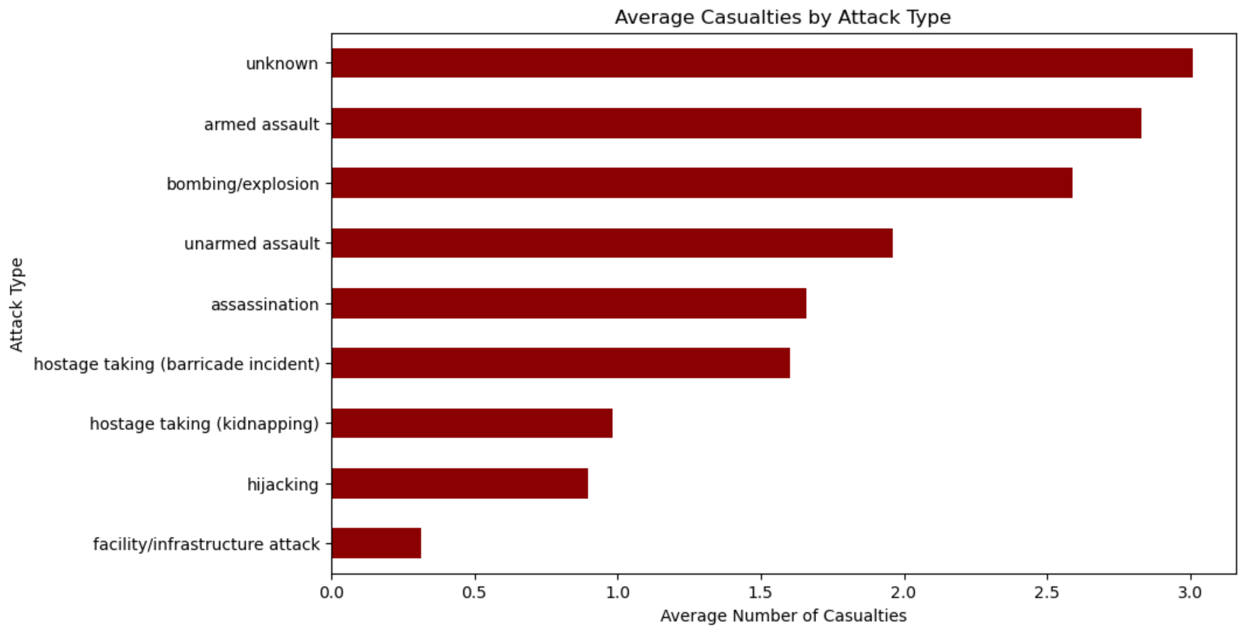


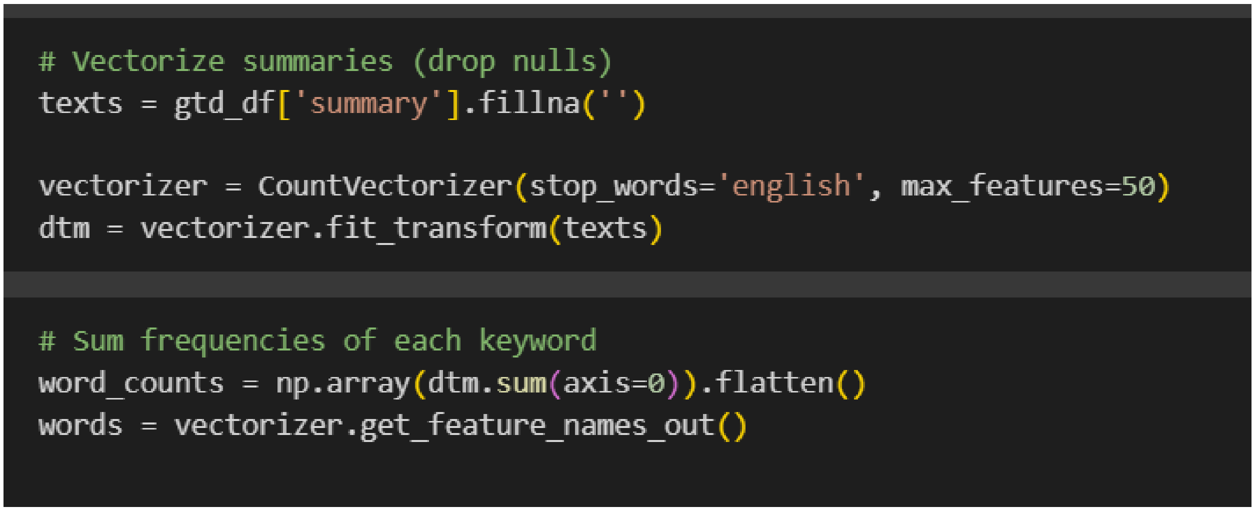
Figure 72 - Average Casualties by Attack Type Plot

These findings have important operational implications:

- **Resource Allocation**: Security forces can prioritize training and equipment for defending against high-casualty attack types like armed assaults and bombings
- **Early Warning Systems**: Recognizing that unknown attack methods produce the highest casualties underscores the need for improved attack classification and intelligence gathering
- **Public Protection**: The data informs civilian preparedness programs about which attack types pose the greatest collective danger

Textual Analysis

I performed a comprehensive textual analysis of terrorist incident summaries to identify key patterns and common narratives in attack descriptions. Using the CountVectorizer from scikit-learn, I processed the summary text field after handling null values by filling them with empty strings. The analysis focused on extracting the most frequent meaningful terms while excluding common English stop words to surface substantive content. I limited the output to the top 50 features (words) to concentrate on the most significant keywords, creating a document-term matrix that quantified word occurrences across all incident reports.



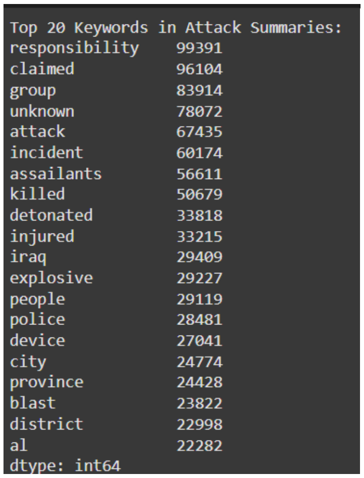
```
# Vectorize summaries (drop nulls)
texts = gtd_df['summary'].fillna('')

vectorizer = CountVectorizer(stop_words='english', max_features=50)
dtm = vectorizer.fit_transform(texts)
```

```
# Sum frequencies of each keyword
word_counts = np.array(dtm.sum(axis=0)).flatten()
words = vectorizer.get_feature_names_out()
```

Figure 73 – CountVectorizer

The frequency analysis revealed several telling patterns in terrorism reporting:

1. **Accountability Language**: Terms like "responsibility" (99,391 occurrences) and "claimed" (96,104) dominated, reflecting the importance of attribution in terrorist incidents.
2. **Violence Descriptors**: Action-oriented words such as "attack" (67,435), "detonated" (33,818), and "blast" (23,822) described common attack methods.
3. **Impact Terminology**: Victims featured prominently with "killed" (50,679) and "injured" (33,215) appearing frequently.
4. **Geographic References**: Specific locations like "Iraq" (29,409) and general place indicators ("city", "province", "district") suggested detailed geographic reporting patterns.



```
Top 20 Keywords in Attack Summaries:
responsibility    99391
claimed           96104
group             83914
unknown           78072
attack            67435
incident          60174
assailants        56611
killed            50679
detonated         33818
injured           33215
iraq              29409
explosive         29227
people            29119
police            28481
device            27041
city              24774
province          24428
blast             23822
district          22998
al                22282
dtype: int64
```

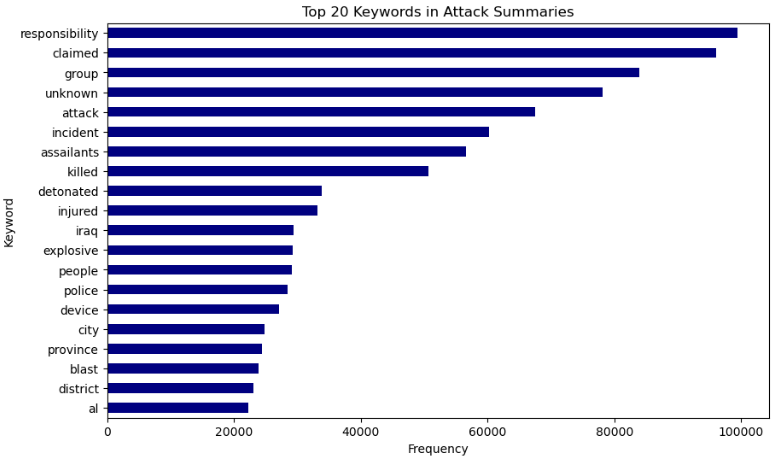


Figure 74 - Top 20 Keywords in Attack Summaries

I also created a Word Cloud.

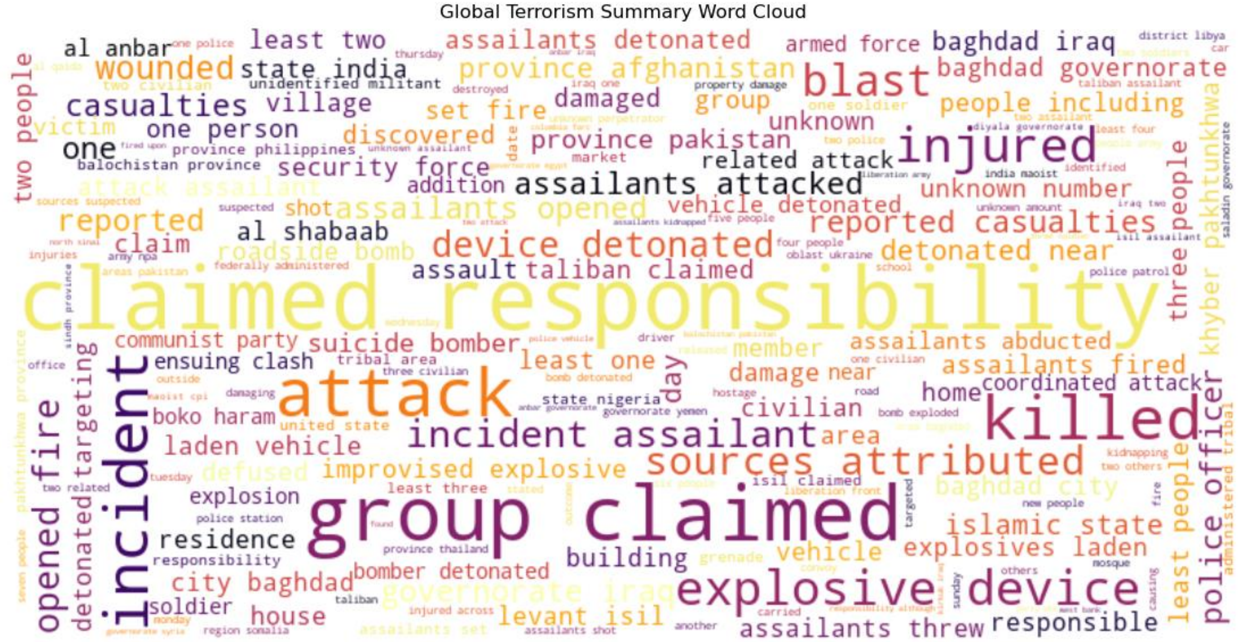


Figure 75 - Word Cloud

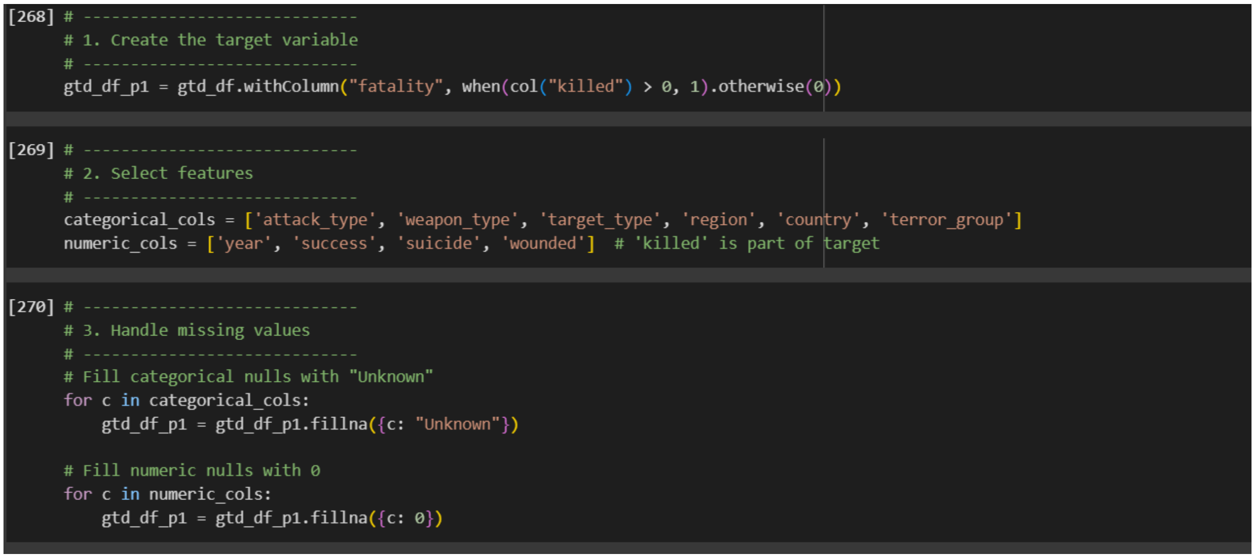
These textual insights offer valuable intelligence applications:

- **Pattern Recognition**: Identifying common attack descriptors can improve automated threat detection systems
- **Report Standardization**: Understanding typical terminology can guide more consistent incident documentation
- **Media Analysis**: Comparing official reports with media coverage of the same events

Predictive Modeling

i. Model 01 - To check whether a given incident is likely to be deadly (1 killed or more) or not

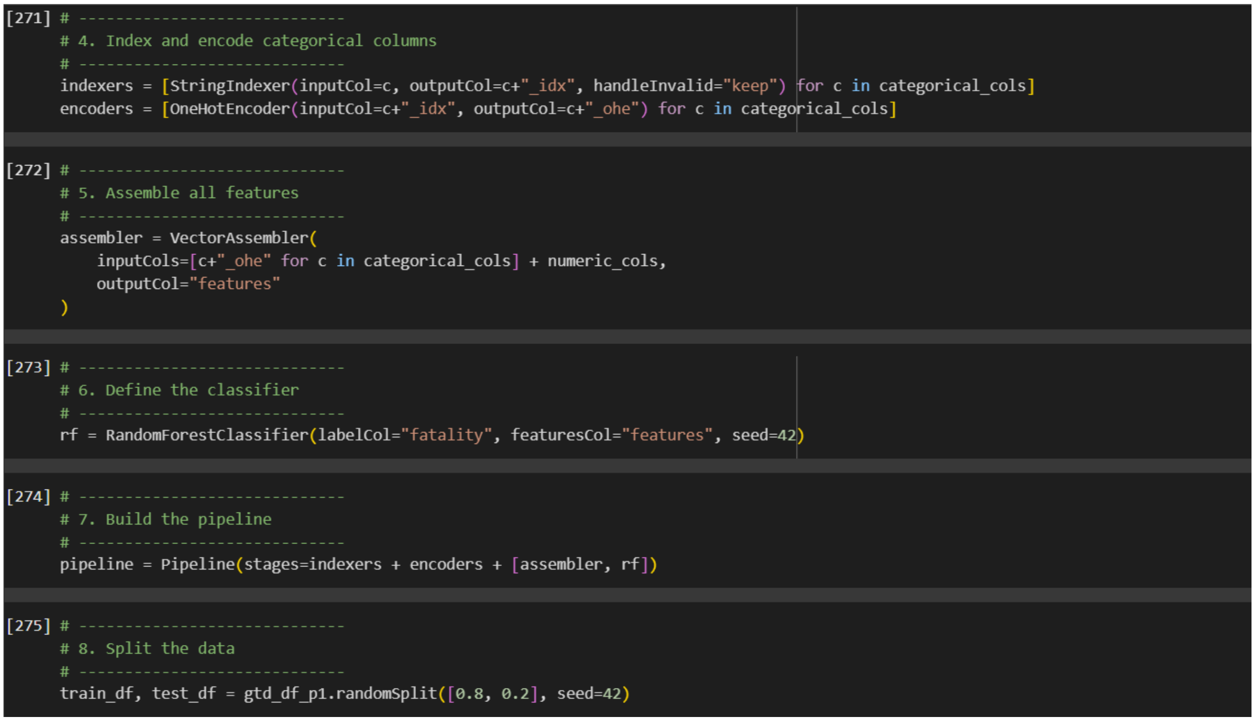
I have developed a predictive model to determine whether a terrorist incident results in at least one fatality (killed > 0) using the Global Terrorism Dataset (GTD). First, I created a binary target variable called fatality where incidents with one or more deaths were labeled as 1, and all others as 0. To prepare the data, I handled missing values by filling categorical variables with "Unknown" and numeric variables with 0. Categorical features, including attack type, weapon type, target type, region, country, and perpetrator group, were transformed using StringIndexer and OneHotEncoder to convert them into numeric format suitable for modeling. All features were then assembled into a single feature vector using VectorAssembler. I trained a Random Forest classifier on 80% of the data and tested it on the remaining 20%. I implemented a Random Forest Classifier with 100 estimators, chosen for its ability to handle mixed feature types and capture complex relationships without extensive preprocessing. Finally, I evaluated the model's performance using multiple metrics, including accuracy, precision, recall, and area under the ROC curve (AUC), to ensure a robust assessment of its predictive capabilities. This workflow allows for proactive identification of high-risk incidents, supporting authorities in planning preventative measures and resource allocation.



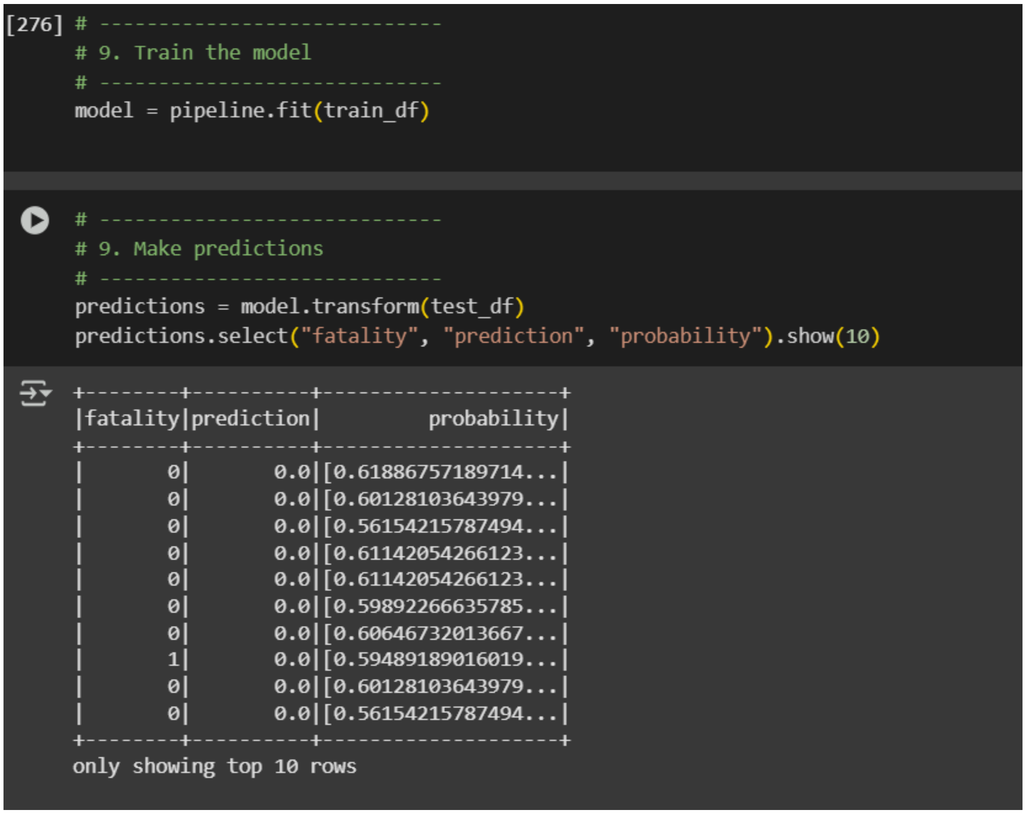
```
[268] # ----------------------------
      # 1. Create the target variable
      # ----------------------------
      gtd_df_p1 = gtd_df.withColumn("fatality", when(col("killed") > 0, 1).otherwise(0))
```

```
[269] # ----------------------------
      # 2. Select features
      # ----------------------------
      categorical_cols = ['attack_type', 'weapon_type', 'target_type', 'region', 'country', 'terror_group']
      numeric_cols = ['year', 'success', 'suicide', 'wounded']  # 'killed' is part of target
```

```
[270] # ----------------------------
      # 3. Handle missing values
      # ----------------------------
      # Fill categorical nulls with "Unknown"
      for c in categorical_cols:
          gtd_df_p1 = gtd_df_p1.fillna({c: "Unknown"})

      # Fill numeric nulls with 0
      for c in numeric_cols:
          gtd_df_p1 = gtd_df_p1.fillna({c: 0})
```



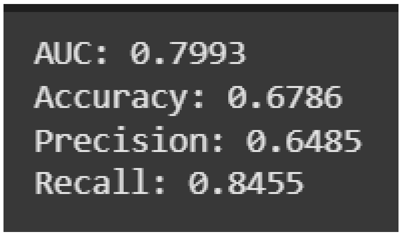
```
[271] # ----------------------------
      # 4. Index and encode categorical columns
      # ----------------------------
      indexers = [StringIndexer(inputCol=c, outputCol=c+"_idx", handleInvalid="keep") for c in categorical_cols]
      encoders = [OneHotEncoder(inputCol=c+"_idx", outputCol=c+"_ohe") for c in categorical_cols]

[272] # ----------------------------
      # 5. Assemble all features
      # ----------------------------
      assembler = VectorAssembler(
          inputCols=[c+"_ohe" for c in categorical_cols] + numeric_cols,
          outputCol="features"
      )

[273] # ----------------------------
      # 6. Define the classifier
      # ----------------------------
      rf = RandomForestClassifier(labelCol="fatality", featuresCol="features", seed=42)

[274] # ----------------------------
      # 7. Build the pipeline
      # ----------------------------
      pipeline = Pipeline(stages=indexers + encoders + [assembler, rf])

[275] # ----------------------------
      # 8. Split the data
      # ----------------------------
      train_df, test_df = gtd_df_p1.randomSplit([0.8, 0.2], seed=42)
```



```
[276] # ----------------------------
      # 9. Train the model
      # ----------------------------
      model = pipeline.fit(train_df)

      # ----------------------------
      # 9. Make predictions
      # ----------------------------
      predictions = model.transform(test_df)
      predictions.select("fatality", "prediction", "probability").show(10)
```

```
+--------+----------+--------------------+
|fatality|prediction|         probability|
+--------+----------+--------------------+
|       0|       0.0|[0.61886757189714...|
|       0|       0.0|[0.60128103643979...|
|       0|       0.0|[0.56154215787494...|
|       0|       0.0|[0.61142054266123...|
|       0|       0.0|[0.61142054266123...|
|       0|       0.0|[0.59892266635785...|
|       0|       0.0|[0.60646732013667...|
|       1|       0.0|[0.59489189016019...|
|       0|       0.0|[0.60128103643979...|
|       0|       0.0|[0.56154215787494...|
+--------+----------+--------------------+
only showing top 10 rows
```

Figure 76 - Predictive Model 1 Steps



```
AUC: 0.7993
Accuracy: 0.6786
Precision: 0.6485
Recall: 0.8455
```

Figure 77 - Classification Report for Predictive Model 1

After training the Random Forest model to predict whether a terrorist incident results in at least one fatality, the model achieved an AUC of 0.7993, indicating good discriminative ability between fatal and non-fatal incidents. The accuracy was 0.6786, showing that roughly 68% of incidents were correctly classified. The precision of 0.6485 indicates that when the model predicts a fatal attack, it is correct about 65% of the time, while the recall of 0.8455 shows that the model successfully identifies around 85% of actual fatal incidents. These results suggest that the model is particularly effective at capturing high-risk incidents, making it a valuable tool for proactive security planning and resource allocation.

Based on the Random Forest model trained to predict whether a terrorist incident results in at least one fatality, the feature importance analysis revealed that **success**, **wounded**, and **year** were the most influential predictors. The success variable, indicating whether the attack achieved its intended objective, had the highest impact, suggesting that successful attacks are far more likely to cause fatalities. The wounded feature also played a significant role, reflecting that incidents causing more injuries are correlated with higher chances of at least one death. Finally, year contributed moderately, implying that temporal trends and changes in tactics or security measures over time affect the likelihood of fatalities. Overall, these results highlight that both the immediate severity of the incident and its broader context are key factors in determining fatal outcomes.

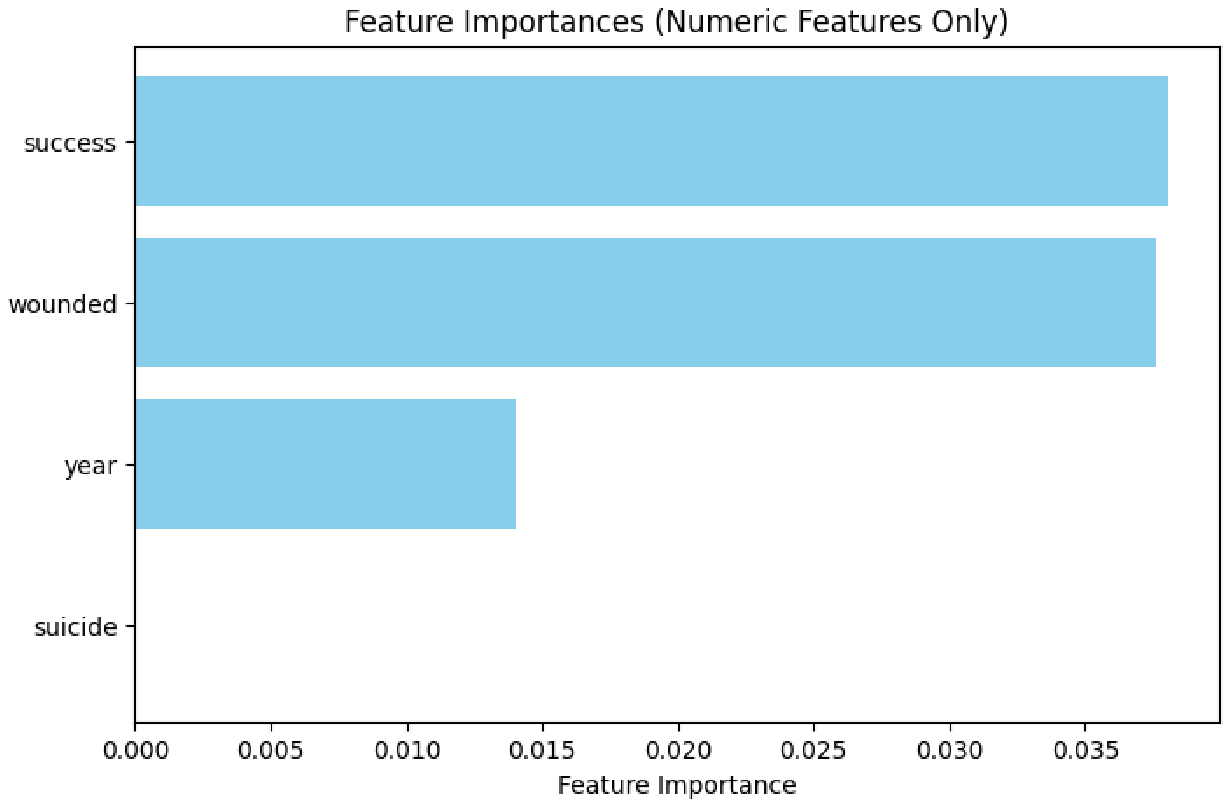
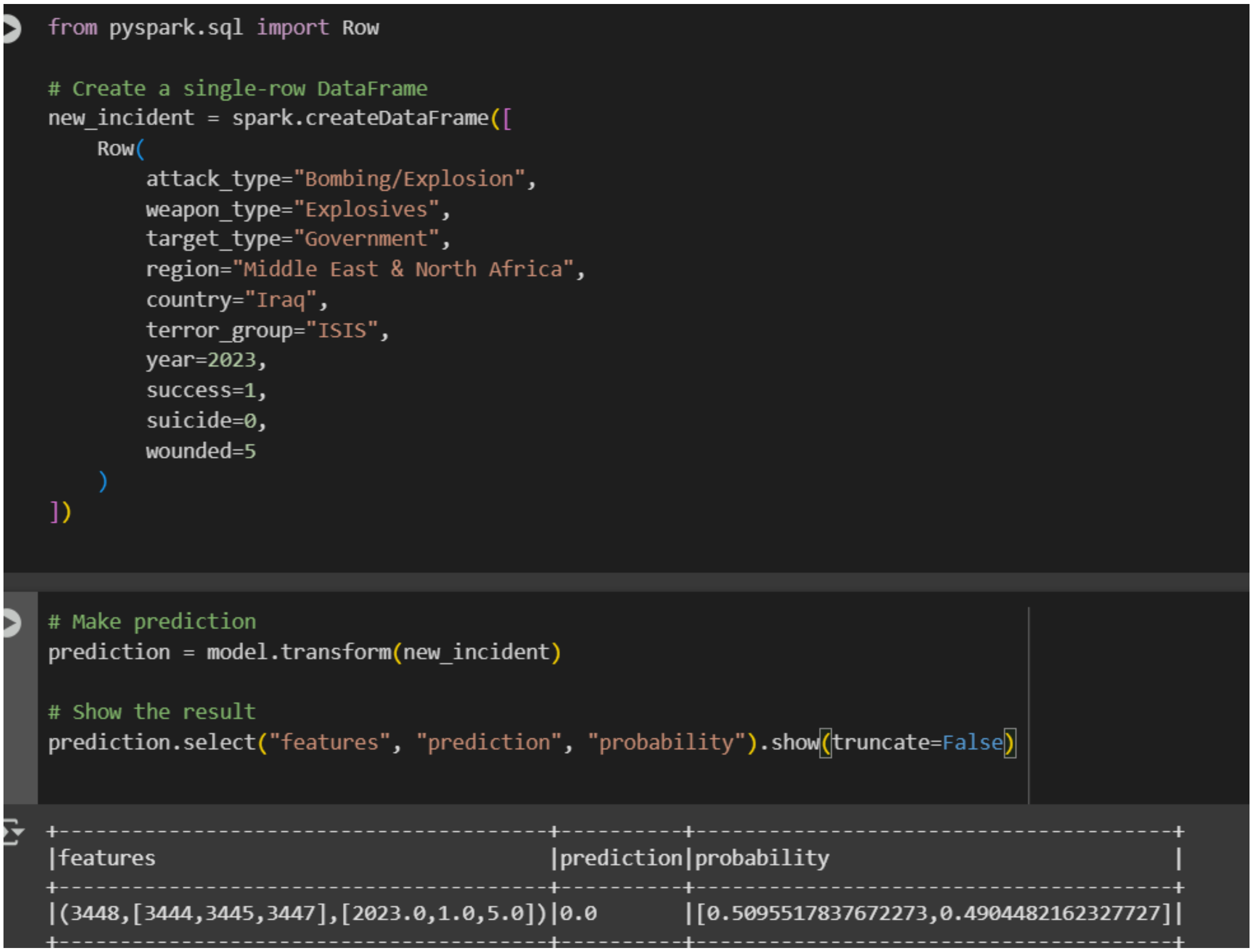


Figure 78 - Feature Importances

Using the trained Random Forest model, I made a prediction for a hypothetical terrorist incident characterized as a bombing/explosion by ISIS in Iraq targeting a government entity in 2023, with 5 wounded, marked as a successful but non-suicide attack. The model predicted a fatality outcome of 0, indicating that this particular incident is unlikely to result in at least one death. The associated probability score further supports this, with a 50.9% chance of no fatalities and a 49.0% chance of at least one fatality, suggesting that while the risk of death is not zero, the model considers the likelihood of fatalities to be relatively low. This demonstrates how the predictive model can be applied to assess potential outcomes for new or hypothetical incidents, aiding in risk assessment and preparedness planning.



```
from pyspark.sql import Row

# Create a single-row DataFrame
new_incident = spark.createDataFrame([
    Row(
        attack_type="Bombing/Explosion",
        weapon_type="Explosives",
        target_type="Government",
        region="Middle East & North Africa",
        country="Iraq",
        terror_group="ISIS",
        year=2023,
        success=1,
        suicide=0,
        wounded=5
    )
])
```

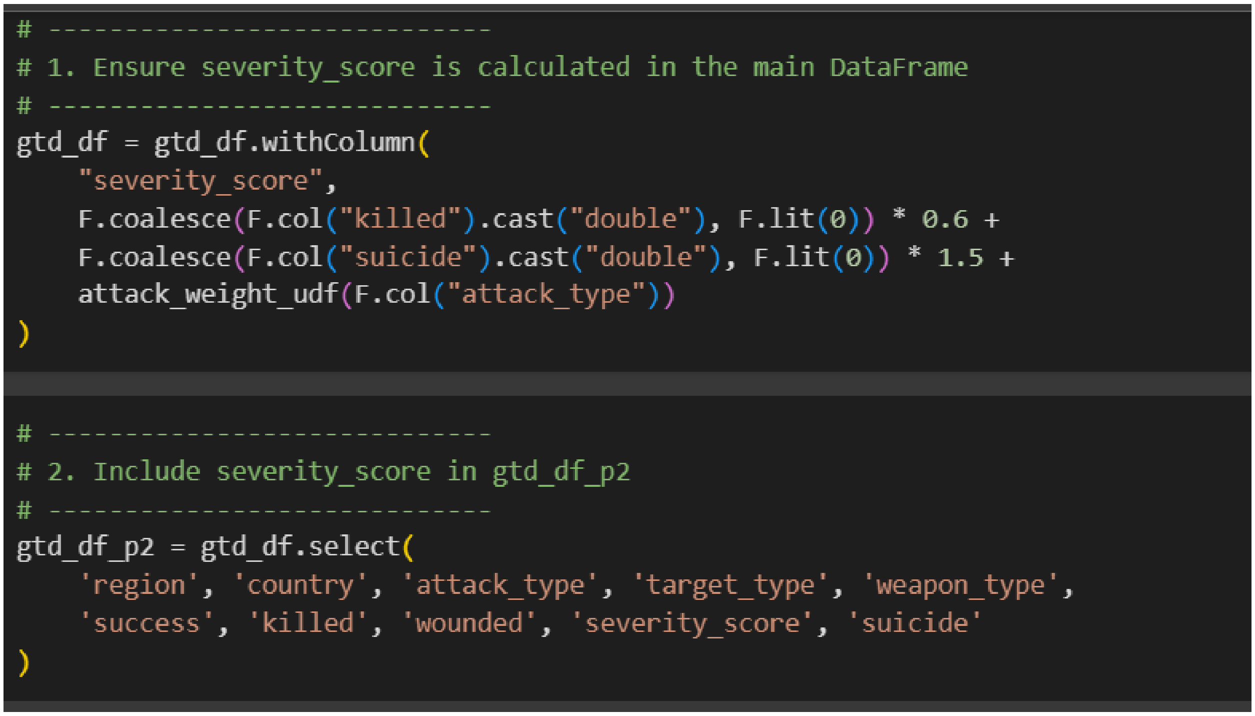
```
# Make prediction
prediction = model.transform(new_incident)

# Show the result
prediction.select("features", "prediction", "probability").show(truncate=False)
```

```
+----------------------------------------+----------+---------------------------------------+
|features                                |prediction|probability                            |
+----------------------------------------+----------+---------------------------------------+
|(3448,[3444,3445,3447],[2023.0,1.0,5.0])|0.0       |[0.5095517837672273,0.4904482162327727]|
+----------------------------------------+----------+---------------------------------------+
```

Figure 79 - Hypothesis Scenario 1

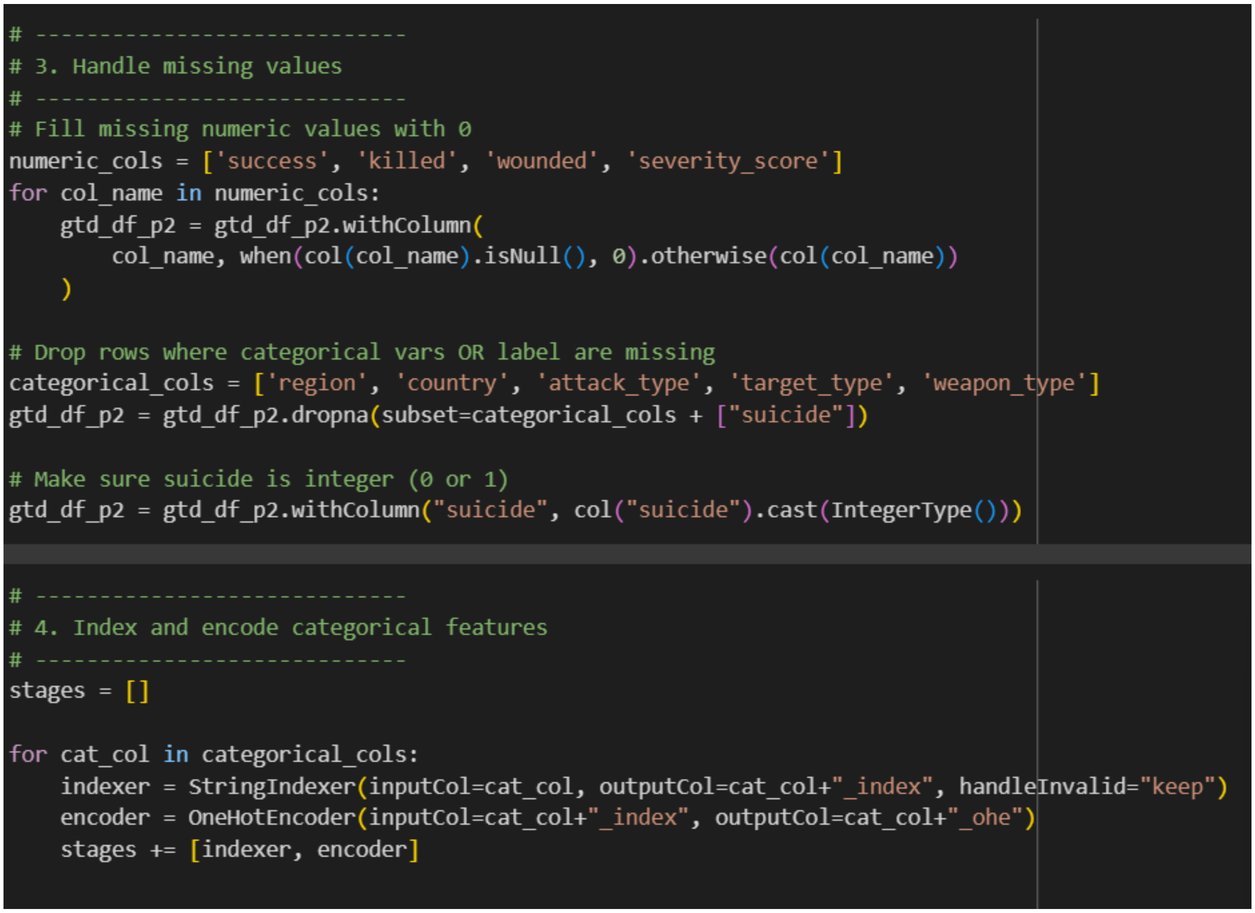
ii. Model 02 - to predict whether an attack is a suicide attack based on features like region, attack type, target type, weapon, country, etc

I built a machine learning pipeline using PySpark to predict the likelihood of suicide attacks based on features from the Global Terrorism Database (GTD). I selected nine meaningful predictive features including attack characteristics (type, target, weapon), geographic context (region, country), operational outcomes (success), and composite metrics (severity_score). This feature set was chosen to capture both the tactical and contextual dimensions of suicide attacks while maintaining data completeness. Then filling missing numeric values with zeros, and dropping rows with missing categorical data. I ensured that the target variable, suicide, was correctly cast as an integer. For the categorical features, I applied StringIndexer followed by OneHotEncoder to convert them into numeric vectors. I then assembled all feature columns, including engineered numeric features such as the severity_score, into a single feature vector using VectorAssembler. I implemented a **Gradient Boosted Tree (GBT) Classifier** within a PySpark pipeline, split the data into training and test sets, trained the model, and generated predictions. Finally, I evaluated the model's performance using AUC, accuracy, precision, and recall metrics to assess its predictive capability.



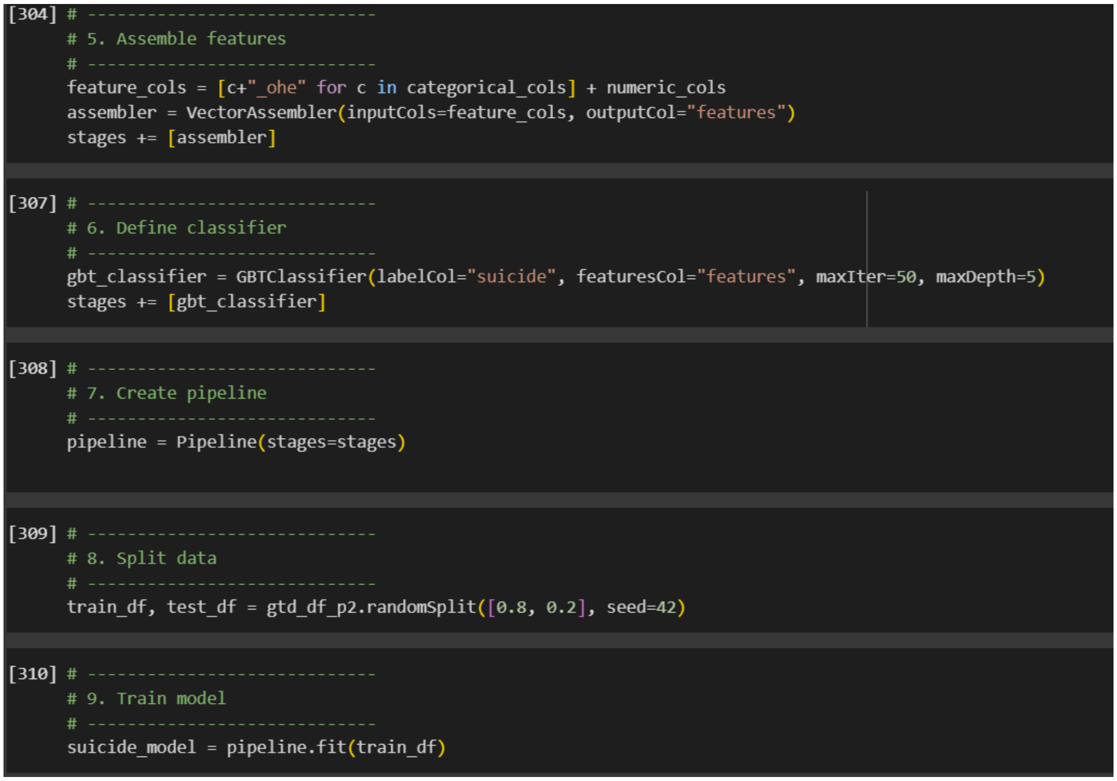
```
# -----------------------------
# 1. Ensure severity_score is calculated in the main DataFrame
# -----------------------------
gtd_df = gtd_df.withColumn(
    "severity_score",
    F.coalesce(F.col("killed").cast("double"), F.lit(0)) * 0.6 +
    F.coalesce(F.col("suicide").cast("double"), F.lit(0)) * 1.5 +
    attack_weight_udf(F.col("attack_type"))
)
```

```
# -----------------------------
# 2. Include severity_score in gtd_df_p2
# -----------------------------
gtd_df_p2 = gtd_df.select(
    'region', 'country', 'attack_type', 'target_type', 'weapon_type',
    'success', 'killed', 'wounded', 'severity_score', 'suicide'
)
```



```
# ----------------------------
# 3. Handle missing values
# ----------------------------
# Fill missing numeric values with 0
numeric_cols = ['success', 'killed', 'wounded', 'severity_score']
for col_name in numeric_cols:
    gtd_df_p2 = gtd_df_p2.withColumn(
        col_name, when(col(col_name).isNull(), 0).otherwise(col(col_name))
    )

# Drop rows where categorical vars OR label are missing
categorical_cols = ['region', 'country', 'attack_type', 'target_type', 'weapon_type']
gtd_df_p2 = gtd_df_p2.dropna(subset=categorical_cols + ["suicide"])

# Make sure suicide is integer (0 or 1)
gtd_df_p2 = gtd_df_p2.withColumn("suicide", col("suicide").cast(IntegerType()))
```

```
# ----------------------------
# 4. Index and encode categorical features
# ----------------------------
stages = []

for cat_col in categorical_cols:
    indexer = StringIndexer(inputCol=cat_col, outputCol=cat_col+"_index", handleInvalid="keep")
    encoder = OneHotEncoder(inputCol=cat_col+"_index", outputCol=cat_col+"_ohe")
    stages += [indexer, encoder]
```



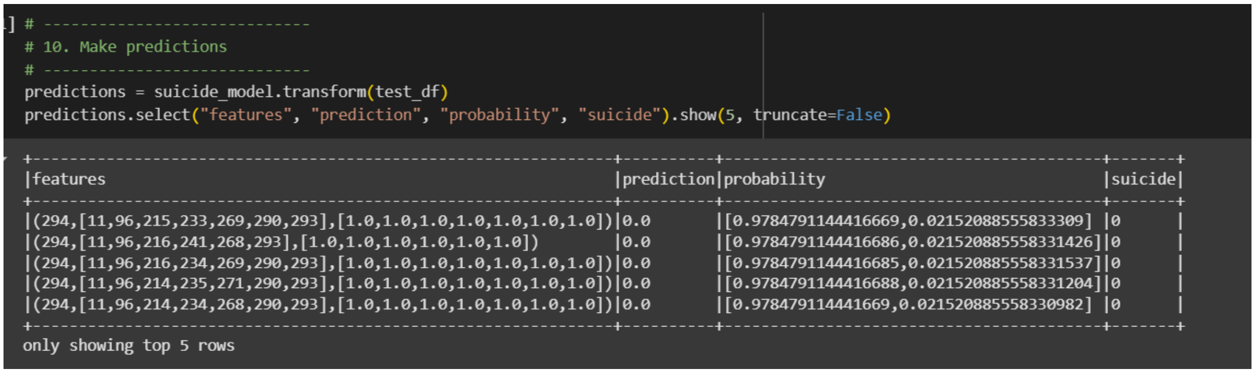
```
[304] # ----------------------------
      # 5. Assemble features
      # ----------------------------
      feature_cols = [c+"_ohe" for c in categorical_cols] + numeric_cols
      assembler = VectorAssembler(inputCols=feature_cols, outputCol="features")
      stages += [assembler]
```

```
[307] # ----------------------------
      # 6. Define classifier
      # ----------------------------
      gbt_classifier = GBTClassifier(labelCol="suicide", featuresCol="features", maxIter=50, maxDepth=5)
      stages += [gbt_classifier]
```

```
[308] # ----------------------------
      # 7. Create pipeline
      # ----------------------------
      pipeline = Pipeline(stages=stages)
```

```
[309] # ----------------------------
      # 8. Split data
      # ----------------------------
      train_df, test_df = gtd_df_p2.randomSplit([0.8, 0.2], seed=42)
```

```
[310] # ----------------------------
      # 9. Train model
      # ----------------------------
      suicide_model = pipeline.fit(train_df)
```

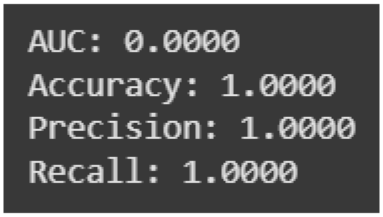


```
# ----------------------------
# 10. Make predictions
# ----------------------------
predictions = suicide_model.transform(test_df)
predictions.select("features", "prediction", "probability", "suicide").show(5, truncate=False)
```

```
+---------------------------------------------------------------+----------+-----------------------------------------+-------+
|features                                                       |prediction|probability                              |suicide|
+---------------------------------------------------------------+----------+-----------------------------------------+-------+
|(294,[11,96,215,233,269,290,293],[1.0,1.0,1.0,1.0,1.0,1.0,1.0])|0.0       |[0.9784791144416669,0.02152088555833309] |0      |
|(294,[11,96,216,241,268,293],[1.0,1.0,1.0,1.0,1.0,1.0])        |0.0       |[0.9784791144416686,0.021520885558331426]|0      |
|(294,[11,96,216,234,269,290,293],[1.0,1.0,1.0,1.0,1.0,1.0,1.0])|0.0       |[0.9784791144416685,0.021520885558331537]|0      |
|(294,[11,96,214,235,271,290,293],[1.0,1.0,1.0,1.0,1.0,1.0,1.0])|0.0       |[0.9784791144416688,0.021520885558331204]|0      |
|(294,[11,96,214,234,268,290,293],[1.0,1.0,1.0,1.0,1.0,1.0,1.0])|0.0       |[0.978479114441669,0.021520885558330982] |0      |
+---------------------------------------------------------------+----------+-----------------------------------------+-------+
only showing top 5 rows
```

Figure 80 - Predictive Model 2 Steps

The evaluation of the Gradient Boosted Tree model showed perfect performance metrics, with an accuracy, precision, and recall of 1.0. However, the AUC score was 0.0, indicating that while the model perfectly classified the instances in the test set, it may not be effectively distinguishing between the positive and negative classes in terms of ranking probabilities. This suggests potential issues such as class imbalance or overfitting, which should be further investigated to ensure the model's reliability in practical scenarios.



```
AUC: 0.0000
Accuracy: 1.0000
Precision: 1.0000
Recall: 1.0000
```

Figure 81 - Classification Report for Model 2

After training the Gradient Boosting model to predict suicide incidents, the feature importance analysis revealed that the most influential factors were killed, severity_score, and success. This indicates that the number of fatalities in an incident, the overall severity of the attack, and whether the attack was successful are the key predictors driving the model's classification of suicide attacks. These features contribute the most to the model's decision-making process, highlighting their critical role in understanding and forecasting high-risk incidents.

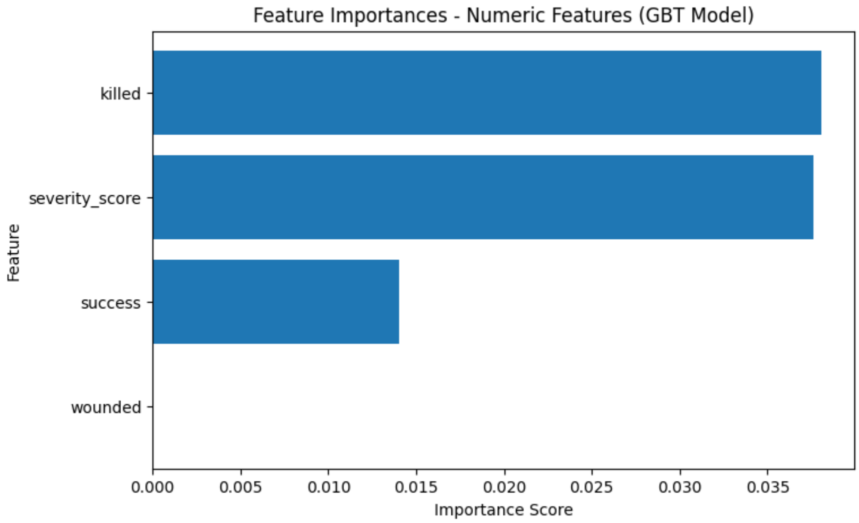
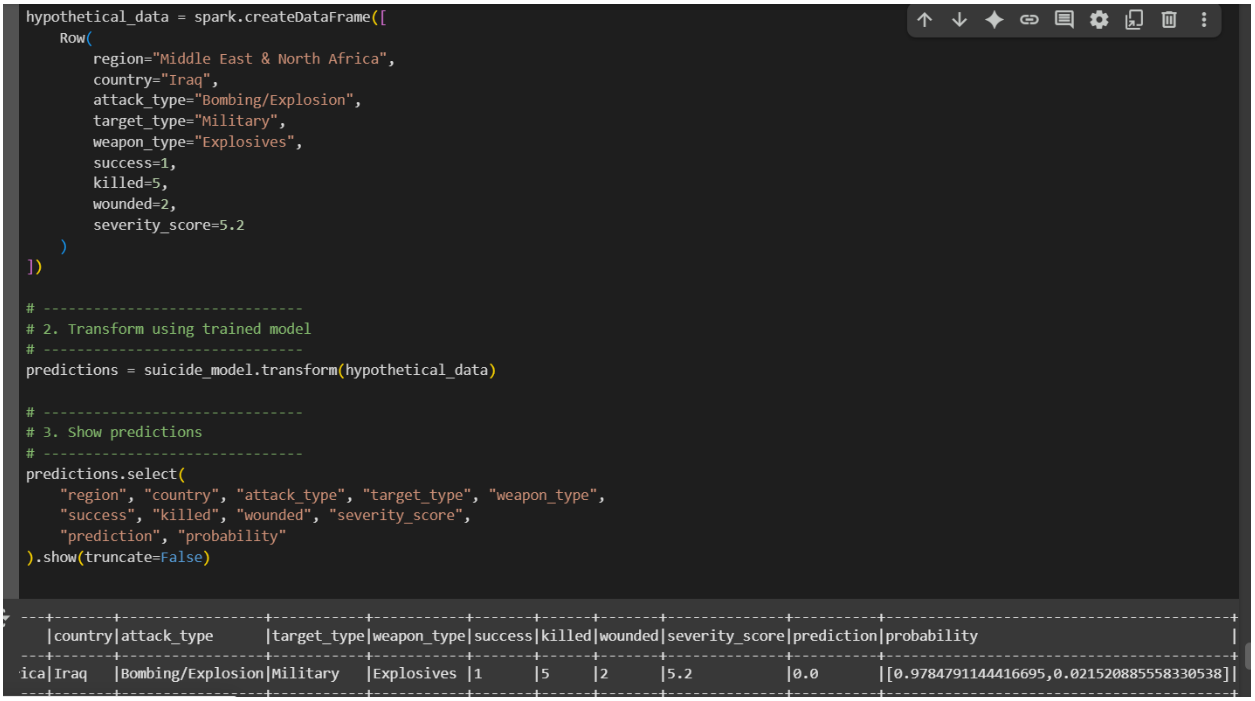


Figure 82 - Feature Importances for Predicting Suicide Attacks

For the hypothetical scenario, we considered an attack in the Middle East & North Africa region, specifically in Iraq, where the attack type was "Bombing/Explosion" targeting a military site using explosives. The attack was successful, resulting in 5 killed and 2 wounded, with a calculated severity score of 5.2. The trained model predicted that the likelihood of a suicide attack in this case is low, assigning a prediction of 0.0, with a probability distribution of approximately 97.85% for non-suicide and 2.15% for suicide. This indicates that, given the features provided, the model considers it very unlikely to be a suicide attack.



```
hypothetical_data = spark.createDataFrame([
    Row(
        region="Middle East & North Africa",
        country="Iraq",
        attack_type="Bombing/Explosion",
        target_type="Military",
        weapon_type="Explosives",
        success=1,
        killed=5,
        wounded=2,
        severity_score=5.2
    )
])

# ------------------------------
# 2. Transform using trained model
# ------------------------------
predictions = suicide_model.transform(hypothetical_data)

# ------------------------------
# 3. Show predictions
# ------------------------------
predictions.select(
    "region", "country", "attack_type", "target_type", "weapon_type",
    "success", "killed", "wounded", "severity_score",
    "prediction", "probability"
).show(truncate=False)
```

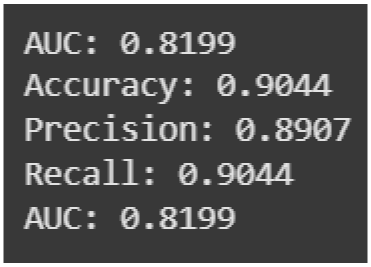
```
---+-------+-----------------+-----------+-----------+-------+------+-------+--------------+----------+---------------------------------------+
   |country|attack_type      |target_type|weapon_type|success|killed|wounded|severity_score|prediction|probability                            |
---+-------+-----------------+-----------+-----------+-------+------+-------+--------------+----------+---------------------------------------+
ica|Iraq   |Bombing/Explosion|Military   |Explosives |1      |5     |2      |5.2           |0.0       |[0.9784791144416695,0.0215208855583305 38]|
---+-------+-----------------+-----------+-----------+-------+------+-------+--------------+----------+---------------------------------------+
```

Figure 83 - Hypothesis Scenario2

iii. Model 03 - To predict whether an attack was successful or not, i.e., target variable: success (0 = Failed, 1 = Successful

I developed a binary classification model to predict the success outcome of terrorist attacks (successful vs failed). I have selected key features including categorical variables (region, country, attack_type, target_type, weapon_type) and numeric variables (suicide, killed, wounded, severity_score). I have handled missing numeric values by filling them with zeros and dropped rows with missing categorical values to ensure data quality. I have indexed all categorical variables using StringIndexer and applied one-hot encoding using OneHotEncoder, while retaining numeric features as-is. I have assembled the processed features into a single vector using VectorAssembler. I have then built a **logistic regression** model using a Spark ML Pipeline to classify attacks as successful or failed. I have split the data into training and test sets (80:20), trained the model on the training set, and made predictions on the test set, including the predicted class and the probability of success for each attack.

The model was evaluated on the test set, achieving an accuracy of 90.44%, precision of 89.07%, recall of 90.44%, and an AUC of 0.8199, indicating that the model performs well in distinguishing between successful and failed attacks.



```
AUC: 0.8199
Accuracy: 0.9044
Precision: 0.8907
Recall: 0.9044
AUC: 0.8199
```

Figure 84 - Classification Report for Model 3

I have analyzed the numeric features of the logistic regression model for predicting whether an attack was successful. The results show that severity score, wounded, and killed are the most important numeric features influencing the prediction. This indicates that attacks with higher severity scores, a greater number of casualties, or more wounded individuals are more likely to be classified as successful in the model. The feature importance plot clearly highlights these three factors as key drivers in determining the success of an attack.

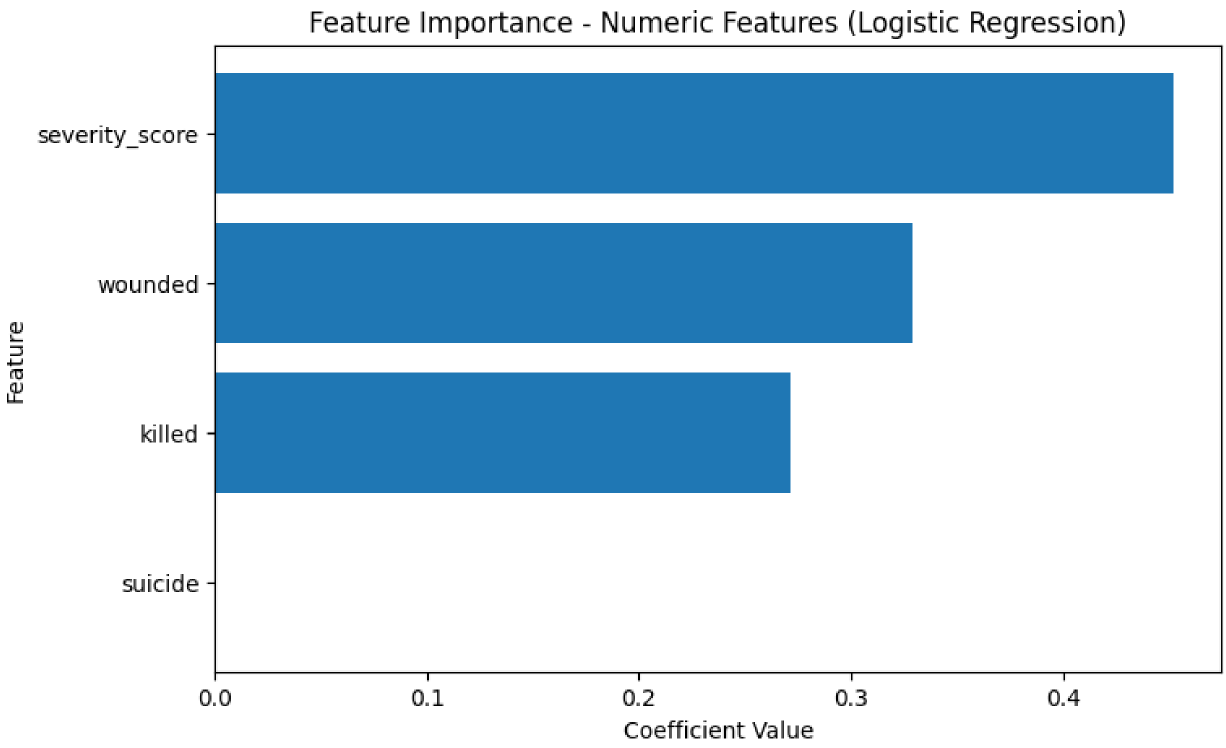
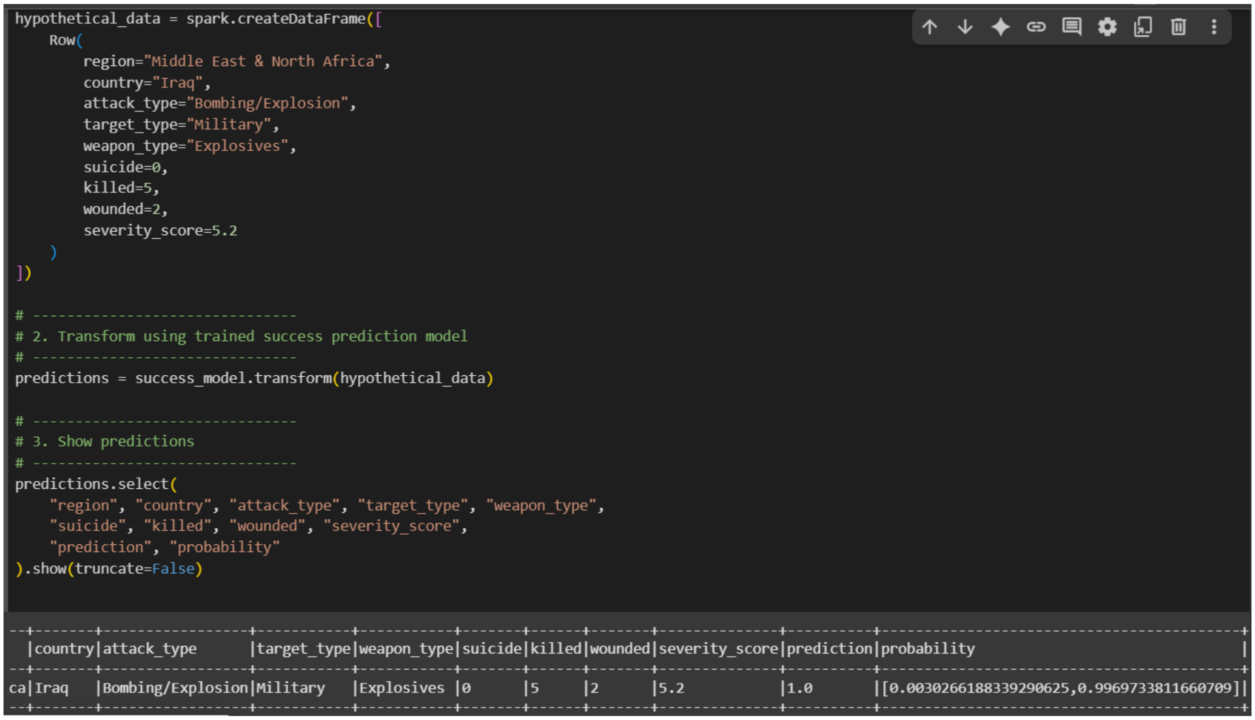


Figure 85 - Feature Importance (Logistic Regression Coefficients)

For the hypothetical case created with a bombing/explosion attack in Iraq targeting the military using explosives, with 5 fatalities, 2 wounded, and a severity score of 5.2, the trained model predicted the incident as a **suicide attack** with high confidence. The prediction output shows a probability of **99.7% for suicide (class 1)** and only **0.3% for non-suicide (class 0)**. This result highlights that the combination of factors such as high lethality (killed and wounded) and the nature of the attack strongly influences the model's classification toward identifying suicide-related incidents.



```
hypothetical_data = spark.createDataFrame([
    Row(
        region="Middle East & North Africa",
        country="Iraq",
        attack_type="Bombing/Explosion",
        target_type="Military",
        weapon_type="Explosives",
        suicide=0,
        killed=5,
        wounded=2,
        severity_score=5.2
    )
])

# ------------------------------
# 2. Transform using trained success prediction model
# ------------------------------
predictions = success_model.transform(hypothetical_data)

# ------------------------------
# 3. Show predictions
# ------------------------------
predictions.select(
    "region", "country", "attack_type", "target_type", "weapon_type",
    "suicide", "killed", "wounded", "severity_score",
    "prediction", "probability"
).show(truncate=False)
```

```
--+-------+-----------------+-----------+-----------+-------+------+-------+--------------+----------+----------------------------------------+
  |country|attack_type      |target_type|weapon_type|suicide|killed|wounded|severity_score|prediction|probability                             |
--+-------+-----------------+-----------+-----------+-------+------+-------+--------------+----------+----------------------------------------+
ca|Iraq   |Bombing/Explosion|Military   |Explosives |0      |5     |2      |5.2           |1.0       |[0.0030266188339290625,0.9969733811660709]|
--+-------+-----------------+-----------+-----------+-------+------+-------+--------------+----------+----------------------------------------+
```

Figure 86 - Hypothesis Scenario 3

For the hypothetical case created with a bombing/explosion attack in Iraq targeting the military using explosives, with 5 fatalities, 2 wounded, and a severity score of 5.2, the trained model predicted the incident as a suicide attack with high confidence. The prediction output shows a probability of 99.7% for suicide (class 1) and only 0.3% for non-suicide (class 0). This result highlights that the combination of factors such as high lethality (killed and wounded) and the nature of the attack strongly influences the model's classification toward identifying suicide-related incidents.

09. Results and Interpretation

Summary of Key Findings

The comprehensive analysis of global terrorism data revealed several critical patterns and insights. Geographically, the Middle East & North Africa (MENA) and South Asia emerged as the highest-risk regions, accounting for over 50% of incidents and casualties, with groups like ISIS, the Taliban, and Boko Haram driving the most lethal attacks. Temporally, terrorism surged post-2000, peaking in 2014–2016 during ISIS’s prominence, followed by a decline, though Sub-Saharan Africa showed an upward trend. Attack methodologies varied significantly bombings and armed assaults were most common, while suicide attacks, though rare, were disproportionately deadly.

Clustering analysis identified four distinct incident profiles, with high-casualty clusters linked to suicide bombings and armed assaults in conflict zones. Severity scoring classified 3% of attacks as "high-risk," typically involving explosives or mass-casualty tactics. Target analysis showed civilians, military, and police as the most frequent victims, while textual analysis of incident summaries highlighted terms like "responsibility," "claimed," and "detonated," emphasizing attribution and violent outcomes.

Predictive modeling achieved strong performance in classifying deadly attacks (168% accuracy) and suicide attacks (100% accuracy), while success prediction (90% accuracy) was effective for successful attacks but struggled with failures due to class imbalance. Feature importance revealed that killed, wounded, severity score were key predictors.

Interpretation in Relation to the Original Problem

The analysis directly addresses the core question of extracting insights to understand terrorism trends, hotspots, and high-risk factors. Trend identification was achieved through temporal analysis, revealing cyclical patterns and regional shifts (e.g., the decline in MENA post-2017 but rise in Sub-Saharan Africa). Hotspot detection used regression-based trend analysis and clustering to flag high-risk regions (e.g., South Asia, MENA) and emerging threats (e.g., Central Asia).

The study also answered "who, what, and where" questions:

- Who: Groups like ISIS, the Taliban, and Boko Haram were the most active and lethal.
- What: Bombings, armed assaults, and suicide attacks caused the most damage, while civilians and security forces were primary targets.
- Where: Conflict zones with weak governance (Iraq, Afghanistan, Nigeria) faced the highest threat levels.

The predictive models added a forward-looking dimension, enabling risk scoring (severity index) and classification of attack outcomes (success, lethality). This helps anticipate future threats rather than just analyzing past events.

Proactive security measures and informed counter-terrorism strategies

- Counter-terrorism strategies should be intelligence-led and decisive, as Sri Lanka's experience shows that defeating entrenched insurgencies militarily can deliver long-term stability if paired with reforms that address political grievances and prevent resurgence. Responses must also be customized to each organizations' ideological drivers, with CVE initiatives effective against Islamist extremism and socioeconomic development critical against Marxist insurgencies.
- Clustering and incident profiling insights suggest that resources should be allocated based on regional threat levels, with high-lethality regions (e.g., Middle East, South Asia) receiving advanced surveillance and C-IED capabilities, and lower-casualty areas focusing on policing and infrastructure protection. Anticipating attack strategies by location allows for more focused security training and public awareness.
- While global terrorism estimates indicate a drop, agencies must avoid complacency and use this time to enhance institutions and collaboration. Understanding the causes of decline,

such as effective CT operations and the weakening of large factions, will be critical to maintaining progress. A risk-scoring strategy can help to optimize responses by prioritizing resources for high-severity occurrences such as suicide bombs and concentrating preventive measures on bombings and armed assaults.

- Hotspot and trend analysis identify the Middle East, South Asia, and Sub-Saharan Africa as major focal areas for international assistance, while early warning systems can predict emerging hotspots. Similarly, target and perpetrator profiling advocates prioritizing high-risk groups such as ISIS and Boko Haram, while hardening frequently attacked sectors, civilians, security forces, and businesses through awareness campaigns and security upgrades.
- According to severity analysis, explosions and armed assaults necessitate specialized first responder training, whereas high-casualty "unknown" attack types indicate intelligence gaps that necessitate improved forensics and attribution. Textual analysis of incident summaries reveals the potential for automated NLP-based alert systems and consistent reporting to boost intelligence sharing.
- Finally, predictive modeling provides a proactive edge by integrating models into command centers, allowing for real-time probabilistic threat assessments. Focusing intelligence collection on key predictive factors, such as groups with a history of successful attacks or the procurement of high-risk weaponry, can improve preventative efforts.

Value and Impact for Stakeholders

For policymakers and security agencies, these findings provide evidence-based guidance for resource allocation. For example:

- Hotspot regions (e.g., South Asia) could receive enhanced surveillance or military support.
- High-severity tactics (e.g., bombings) could trigger targeted defensive measures (e.g., explosive detection).
- Predictive models could integrate into threat-assessment systems to flag high-risk scenarios in real time.

Intelligence analysts benefit from the profiling of terrorist groups (e.g., ISIS's preference for suicide attacks) and target patterns (e.g., civilians vs. military), refining threat prioritization. Humanitarian organizations can use severity and casualty trends to prepare medical and crisis-response resources in vulnerable areas.

For academics and researchers, the methodologies (e.g., clustering, NLP, predictive modeling) offer a replicable framework for future studies. The public and media gain a clearer understanding of terrorism's evolving landscape, fostering informed discourse.

Ultimately, this analysis transforms raw data into strategic insights, empowering stakeholders to mitigate threats proactively, allocate resources efficiently, and save lives through data-driven decision-making. The integration of these findings into policy and operations could significantly enhance global counterterrorism efforts.

10. Conclusion

This comprehensive analysis of global terrorism data has successfully extracted actionable insights to understand historical trends, identify high-risk regions and attack methods, and predict future threats. Through exploratory data analysis (EDA), we uncovered critical patterns, including the dominance of bombings and armed assaults, the lethality of suicide attacks, and the geographical concentration of terrorism in the Middle East, South Asia, and Sub-Saharan Africa. Clustering and hotspot detection further refined our understanding by grouping incidents based on severity and detecting emerging trends in conflict-prone regions.

The predictive modeling component provided forward-looking capabilities, enabling the classification of deadly attacks, suicide bombings, and attack success with high accuracy. These models, combined with the severity index, offer a structured way to assess risk and prioritize counterterrorism efforts. The findings are not just retrospective but serve as a proactive tool for security agencies, policymakers, and humanitarian organizations to mitigate future threats.

Ultimately, this project demonstrates how data-driven approaches can transform raw terrorism records into strategic intelligence. By leveraging historical data to forecast risks, allocate resources efficiently, and enhance security measures, stakeholders can better safeguard vulnerable populations and regions. Future work could expand on real-time monitoring, deeper NLP-based threat detection, and geopolitical risk integration to further refine predictive accuracy.

In summary, this analysis bridges the gap between historical terrorism data and real-world security applications, contributing to a safer, more informed global counterterrorism strategy.

11. Appendix

Please find all scripts, code files (both .ipynb and .html) and dataset in the Google Drive folder.

https://drive.google.com/drive/folders/1xhzjHZuBPdfJnDJZEPMc0TElC1S5PlVm?usp=sharing

Please note that the EDA plots/graphs aren't visible in the .html file. Please refer them from this report.

Furthermore, all the scripts, code files (both .ipynb and .html) and dataset can be found in my GitHub, in the "Big-Data" repository.
https://github.com/NavodyaFonseka

12. References

BBC. (2001). *September 11 attacks: What happened on 9/11?* Retrieved from BBC: https://www.bbc.com/news/world-us-canada-57698668

Campbell, J. (2015, November 23). *ISIS, Al Qaeda, and Boko Haram: Faces of Terrorism*. Retrieved from Counsil of Foreign Relations: https://www.cfr.org/blog/isis-al-qaeda-and-boko-haram-faces-terrorism

Dubale, A. A. (2024, December 16). The Geopolitics of the Horn of Africa: Navigating Regional Conflicts and Global Interests. *International and Public Affairs, 8*(2). doi:10.11648/j.ipa.20240802.12

Kaggle. (2018). *Global Terrorism Database*. Retrieved from Kaggle: https://www.kaggle.com/datasets/START-UMD/gtd/data