Python Advanced

Global Terrorism Project

Youssef Shaaban Sayed Mohamed

Introduction

In this project, we will prepare and analyze The Global Terrorism Dataset. And extract some interesting insights from it. After that, we will see how to use the Dask library and compare it with the Pandas library. And if it affects the performance of the usage of the memory.

Dataset

The Global Terrorism Database (GTD) is a publicly accessible database that catalogs information on terrorist attacks worldwide spanning from 1970 to 2017. It provides comprehensive data on both domestic and international incidents, encompassing over 180,000 attacks during this period. It has 135 columns and 181691 rows.

Data preprocessing and cleansing

Exploration:

The data contains a lot of columns that have nulls as we can see in figure 1. In addition to that, we will not use all columns for analysis so we chose some of them which are:

[Year, Month, Day, Country, Region, City, State, Latitude, Longitude, Success, Suicide, AttackType, Killed, Wounded, Casualties, Target, Group, Target_type, Weapon_type]

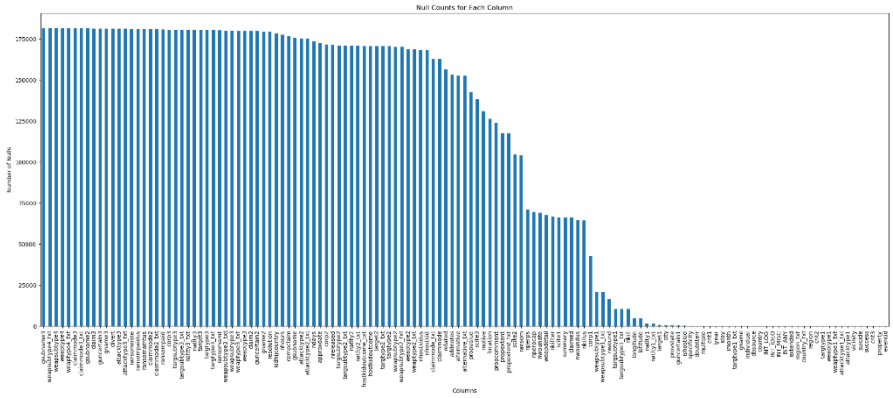

Figure 1 Number of nulls in each column.

Cleaning:

There are some nulls in the selected columns. Here is how we treated them:

- For the City, State, and Target col. we replaced the nulls with “Unknown”.
- For the Killed, Wounded, and Casualties col. we replaced the nulls with 0.
- For the latitude and longitude, we calculate the mean latitude and longitude for each region where known. Then, fill in missing latitude and longitude based on region means.

After that, we checked the duplicates and it was 0 duplicates found.

Data Analysis

We perform basic statistical analysis using Numpy to summarize the data. We calculate the mean, median, and standard deviation for Killed, Wounded, and Casualties columns.

Killed:

- Mean = 2.27: On average, 2.27 people were killed per terrorist attack.
- Median = 0.0: The middle value of the number of people killed is 0, indicating that many attacks result in no fatalities.
- Standard Deviation = 11.23: The number of people killed varies significantly across attacks, with a standard deviation of 11.23.

****

Wounded:

- Mean = 2.88: On average, 2.88 people were wounded per terrorist attack.
- Median = 0.0: The middle value of the number of people wounded is 0, indicating that many attacks result in no injuries.
- Standard Deviation = 34.31: The number of people wounded varies significantly across attacks, with a standard deviation of 34.31.

****

Casualties:

- Mean = 4.80: On average, 4.80 people (killed or wounded) were affected per terrorist attack.
- Median = 1.0: The middle value of the number of casualties is 1.0, indicating that in many attacks, either 1 person was killed or wounded.
- Standard Deviation = 40.10: The number of casualties (killed or wounded) varies significantly across attacks, with a standard deviation of 40.10

After that we want to see the most frequent values in ['Country', 'Region', 'City', 'State', 'AttackType', 'Target', 'Group', 'Target_type', 'Weapon_type']

- **Iraq** is the country where the highest number of terrorist attacks have been recorded in the dataset.
- **The Middle East & North Africa** region has experienced the highest frequency of terrorist attacks based on the dataset.
- **Baghdad** is the city where the highest number of terrorist attacks have occurred according to the dataset.
- **Baghdad** is also the state (or province) within Iraq where the highest number of terrorist attacks have been reported.
- **Bombing/Explosion** is the most frequently used method of attack in the dataset, indicating its prevalence as a tactic among terrorist groups.
- **Civilians** are the most common target of terrorist attacks, highlighting the impact on non-combatant populations.
- **The Taliban group** is identified as the most frequent perpetrator of terrorist attacks based on the dataset.
- **Private Citizens & Property** are the most commonly targeted types, indicating attacks aimed at individuals and civilian infrastructure.

- **Explosives** are the most commonly used weapon type in terrorist attacks, underscoring their widespread use as a method of attack.

We now will use Pandas to some other analysis. And here some insights for doing some aggregations.

- **2014** was the year that had the most number of attacks with 16903 attacks.
- The **Taliban group** had the most number of the cities they attacked around 2075 city.
- The group that had the most a number of Casualties 58223 was **Islamic State of Iraq and the Levant (ISIL) group.**
- **Islamic State of Iraq and the Levant (ISIL) group** was the most group used **Suicide** as a way for the attack.
- **Islamic State of Iraq and the Levant (ISIL) group** used to attack Private Citizens & Property, whereas **Taliban group** used to attack Police.

Data Visualizations

We visualized some insights for better understanding the trends.

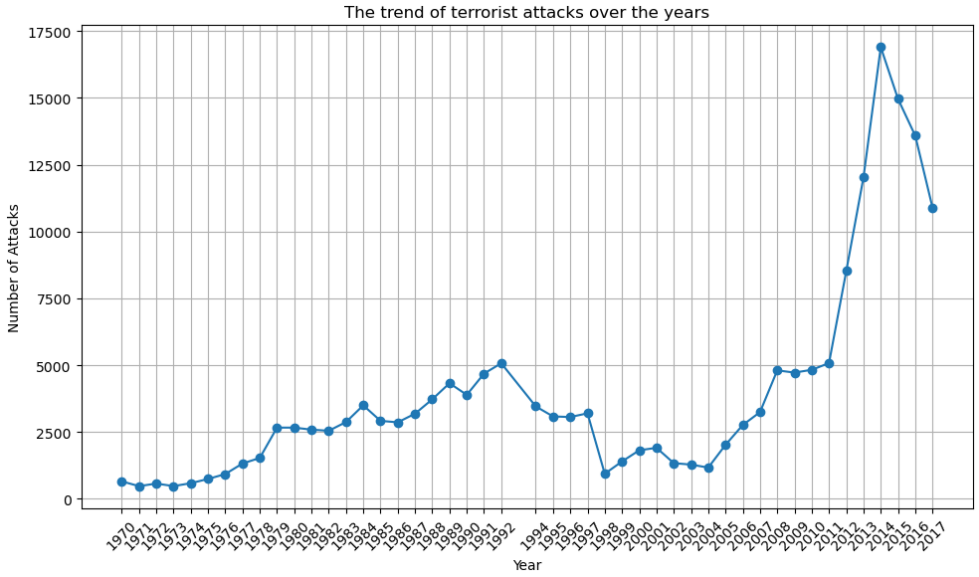

Figure 2 Trend of terrorist attacks over the years

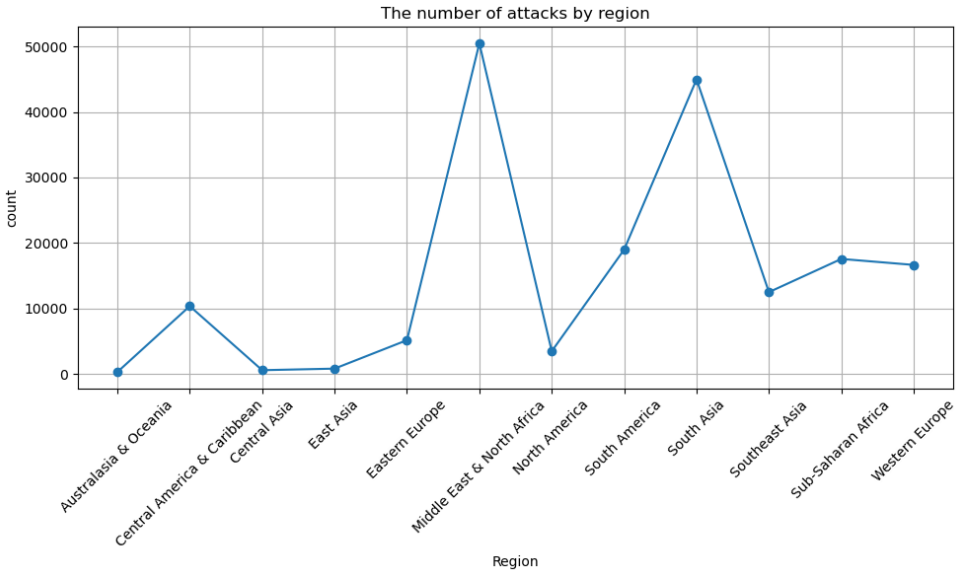

Figure 3 The number of attacks by region

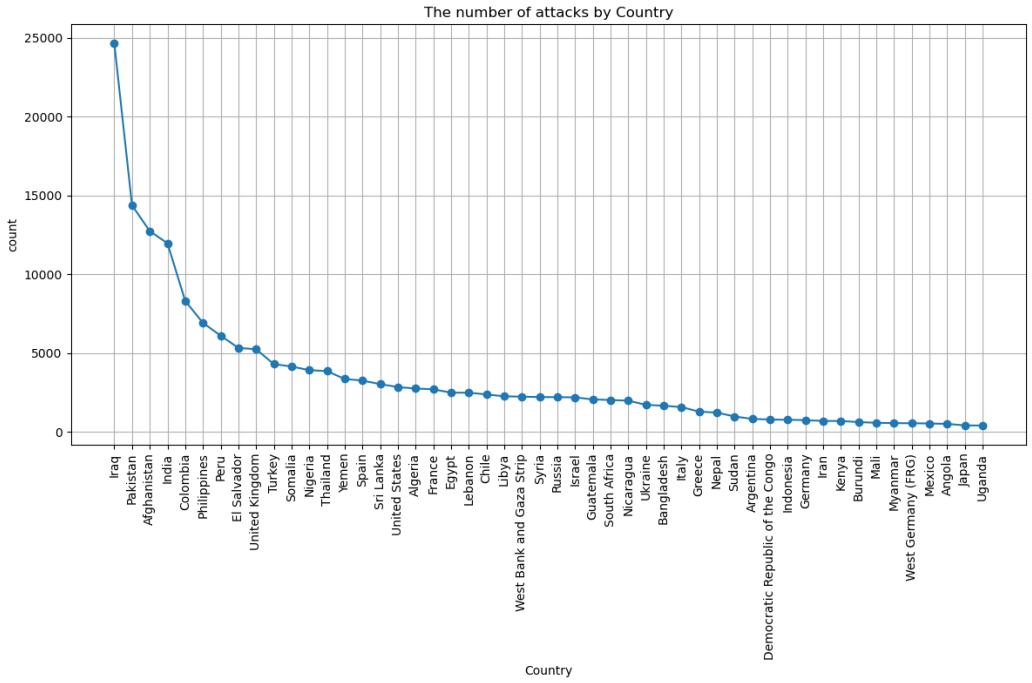

Figure 4 The number of attacks by Country

We can see here there are no strong correlations except for the Casualties and killed and wounded. This does make sense as the Casualties are the summation of both killed and wounded.

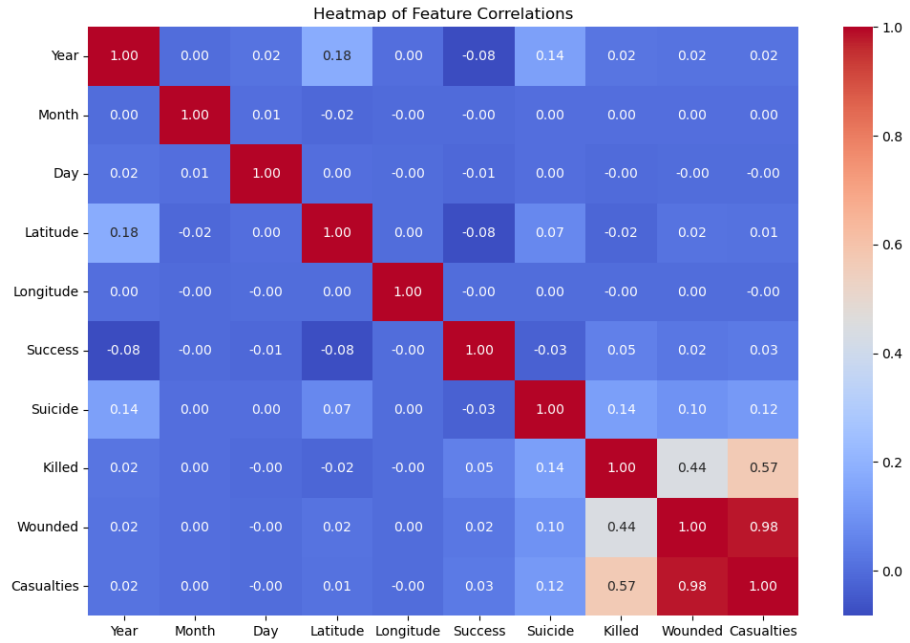

Figure 5 Heatmap of Feature Correlations

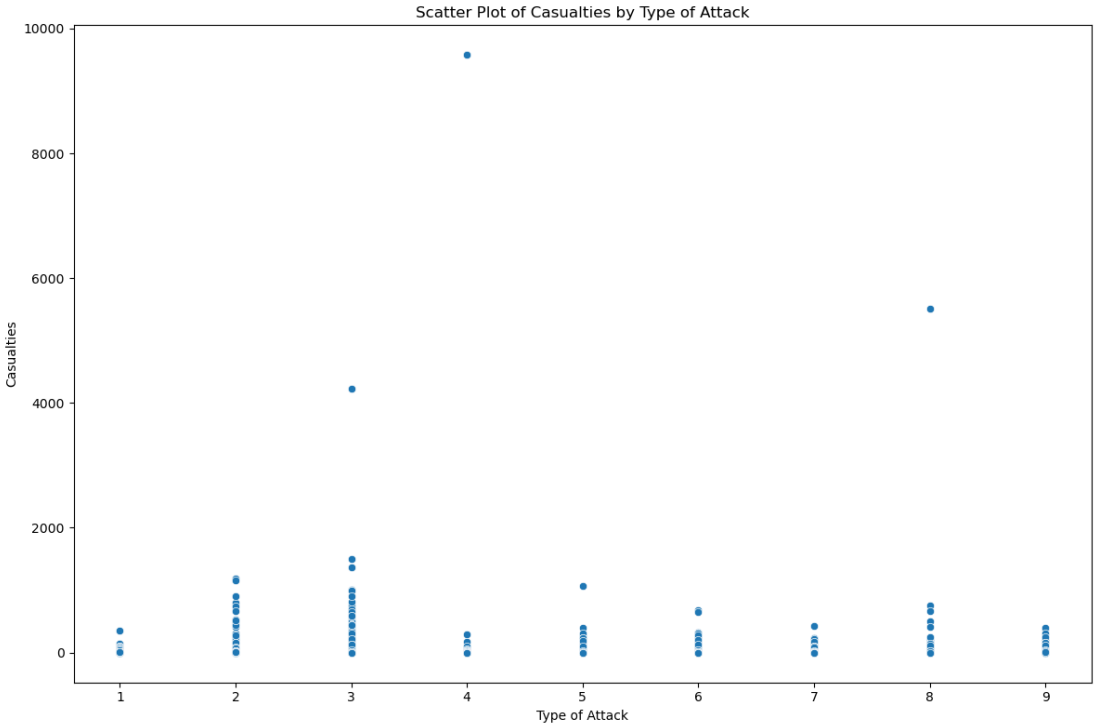

Figure 6 Scatter Plot of Casualties by Type of Attack

Performance Comparison with Dask

After testing Dask with 5 partitions, we found that NumPy outperformed Dask for our dataset, despite similar memory usage between the two. Surprisingly, increasing the number of partitions in Dask did not improve performance; instead, it slowed down processing. This could be attributed to our dataset's relatively small size, where Dask's overhead in managing partitions outweighed any potential gains from parallel processing. Dask is typically more advantageous for larger datasets that benefit from distributed computing across multiple cores or machines to achieve significant performance enhancements.

Challenges During Analysis

One of the primary challenges encountered during the analysis was navigating the extensive number of columns within the dataset and determining which ones were most pertinent for analysis. This process required significant time and effort to understand the information contained in each column and prioritize them based on relevance to the analysis goals.

Another significant challenge involved handling missing data (nulls) effectively to ensure that it did not bias or distort the analysis results. Addressing null values appropriately was crucial to maintaining the integrity and accuracy of the findings.