



Analyzing Terrorism Trends

*Pavithra Shetty
Kapil Singla
Tanmay Rewari
Vinay Khandelwal*



Abstract

Demystifying Terror: Unveiling Patterns and Predicting Outcomes with Machine Learning

This project delves into the vast Global Terrorism Database (GTD), seeking to unveil hidden patterns and trends within terrorist attacks. Our goal is to empower informed decision-making, potentially saving lives in the face of these threats. We've also developed Machine Learning algorithms capable of predicting the success of terrorist attacks. This predictive ability empowers governments and anti-terrorism organizations to tailor their response, recognizing that different attacks require varying degrees of counteraction.

Unveiling the Landscape of Terrorism

The GTD, spanning over five decades, underwent meticulous cleaning and pre-processing to ensure its suitability for training our models. We acknowledge and address the inherent bias within the data to prevent models that are mere statistical illusions, appearing effective on paper but lacking real predictive power. This involves employing Oversampling techniques and utilizing alternative performance metrics that go beyond the traditional ones often utilized in Machine Learning.

Our Machine Learning models analyze a vast array of factors, including target type, attack type, location, and perpetrators, to assess the likelihood of an attack's success. This predictive power allows governments and organizations to allocate resources more effectively, potentially preventing devastating attacks and saving lives.

While the majority of our models fall under the umbrella of Supervised Learning, we also incorporate a K-Means Clustering algorithm into our analytical toolbox. This allows us to identify potential subgroups within the data, uncovering hidden patterns and relationships that may not be readily apparent with traditional methods. Although the binary nature of our target variable

(success or failure of an attack) may limit the formation of distinct clusters, K-Means Clustering still provides valuable insights into the complex landscape of terrorism.

A Holistic Approach to Counterterrorism:

This project showcases a multifaceted approach to tackling a large and inherently biased dataset. We emphasize the importance of nuanced modeling and evaluation techniques for effective counterterrorism efforts, ultimately striving to make a significant contribution in the fight against global terrorism.

By combining data analysis, machine learning, and sophisticated algorithms, we aim to provide a more comprehensive understanding of terrorism and empower organizations to make informed decisions in the face of this complex and evolving threat.

Introduction

Terrorism has cast a long shadow over the global landscape for decades, leaving a trail of devastation and fear in its wake. Understanding the intricacies of this event has become increasingly crucial, necessitating a comprehensive approach to data collection and analysis. While numerous entities have undertaken efforts to document and categorize terrorist attacks, many operate within the confines of privacy restrictions, limiting the accessibility and scope of their data.

To overcome these limitations and facilitate our research, we turned to the invaluable resource of the Global Terrorism Database (GTD). This open-source repository, maintained by the National Consortium for the Study of Terrorism and Responses to Terrorism (START) at the University of Maryland, provides a wealth of information on terrorist incidents worldwide. Spanning from 1970 to 2021, the GTD boasts over 200,000 entries, each meticulously documented with up to 135 separate attributes. This comprehensive dataset encompasses details such as date, location, target type, weapon type, casualties, and the responsible group or individual.

The GTD's commitment to transparency and accessibility extends beyond the sheer volume of data it provides. START also offers valuable insights into its codebook, data collection methods, and methodology, enabling researchers to gain a deeper understanding of the data's strengths and limitations. This transparency is crucial for ensuring the accuracy and reliability of the research conducted using the GTD, ultimately leading to more informed and effective counter-terrorism strategies.

By leveraging the GTD's extensive data and START's meticulous methodology, our project aims to shed light on the complex factors influencing terrorist attacks and their success. Through careful analysis and the application of machine learning techniques, we hope to identify key variables associated with attack outcomes, ultimately enabling the development of predictive models capable of mitigating future threats and safeguarding lives.

Related Work

In addition to the GTD, we also referred to other sources to gain a deeper understanding of current terrorism trends and insights into specific issues related to different countries. The U.S. Department of State and the National Counter Terrorism Center proved to be invaluable resources. These sources provide country-specific information on terrorist issues, shedding light on evolving trends. We believe that this background research will enhance our understanding of various aspects of terrorism, including the types of attacks, the regional organizations responsible for these acts, and the potential for their success.

By tapping into these resources, we aim to gather information that will be reflected in our dataset. Key topics discussed in our research include countries experiencing turmoil, such as Syria and Somalia, as well as insights into the preferences and activities of organizations like Al-Qaeda, al-Shabaab, and Hezbollah, which are on the rise. These elements contribute to a comprehensive understanding of the landscape of terrorism, enabling us to make informed and data-driven analyses.

Models

K-Nearest Neighbors (KNN):

Description:

K-Nearest Neighbors (KNN) is a non-parametric, lazy learning classifier used for supervised learning tasks. It relies on the proximity or similarity between data points to make predictions or classifications. KNN predicts the class of a given instance by considering the k nearest neighbors.

Workflow:

- Compute the distance between the target instance and all data points in the dataset.
- Identify the k nearest neighbors based on distance.
- Assign the class label by majority voting among the k neighbors.

Tuning:

Grid search was employed to tune the hyperparameters of the KNN model for optimal performance. The user-defined parameter k was varied to find the most suitable value.

Support Vector Machine (SVM):

Description:

Support Vector Machine (SVM) is a supervised learning algorithm used for classification and regression tasks. SVM finds the optimal hyperplane that separates data points into different classes, maximizing the margin between classes. The Soft Optimal Margin Classifier allows for some misclassification (slack variables ξ) and includes a balancing parameter C .

Workflow:

- Formulate the equation of the hyperplane $z = w^T(x) + b$ for separating classes.

- Define the Soft Optimal Margin Classifier with slack variables ξ_i and parameter C .
- Formulate the Lagrangian Function and the Dual Function.
- Solve the dual problem to find optimal dual variables α^* .
- Obtain primal optimal values for w^* , b^* , and ξ^* .

Principal Component Analysis (PCA):

Description:

Principal Component Analysis (PCA) is an unsupervised technique used for dimensionality reduction. It aims to transform high-dimensional data into a lower-dimensional representation while preserving the maximum variance. PCA identifies the principal components (eigenvectors) associated with the highest eigenvalues.

Workflow:

- Correlate high-dimensional data.
- Center the data to zero mean.
- Compute the covariance matrix.
- Calculate eigenvectors and eigenvalues.
- Select eigenvectors of lower dimension with the highest eigenvalues.
- Project data points onto selected eigenvectors.

Gradient Boosting Machine (GBM):

Description:

Gradient Boosting Machine (GBM) is an ensemble learning technique that builds a strong predictive model by combining the outputs of multiple weak learners, often decision trees.

GBM sequentially fits new models to the residual errors of the existing model, thereby reducing prediction errors.

Workflow:

- Build an initial weak learner (usually a shallow decision tree).
- Calculate residuals between predictions and actual values.
- Fit a new weak learner to the residuals.
- Combine predictions of all weak learners.
- Update residuals and repeat steps 3-4 for a predefined number of iterations.
- Obtain the final boosted model with improved predictive accuracy.

Tuning:

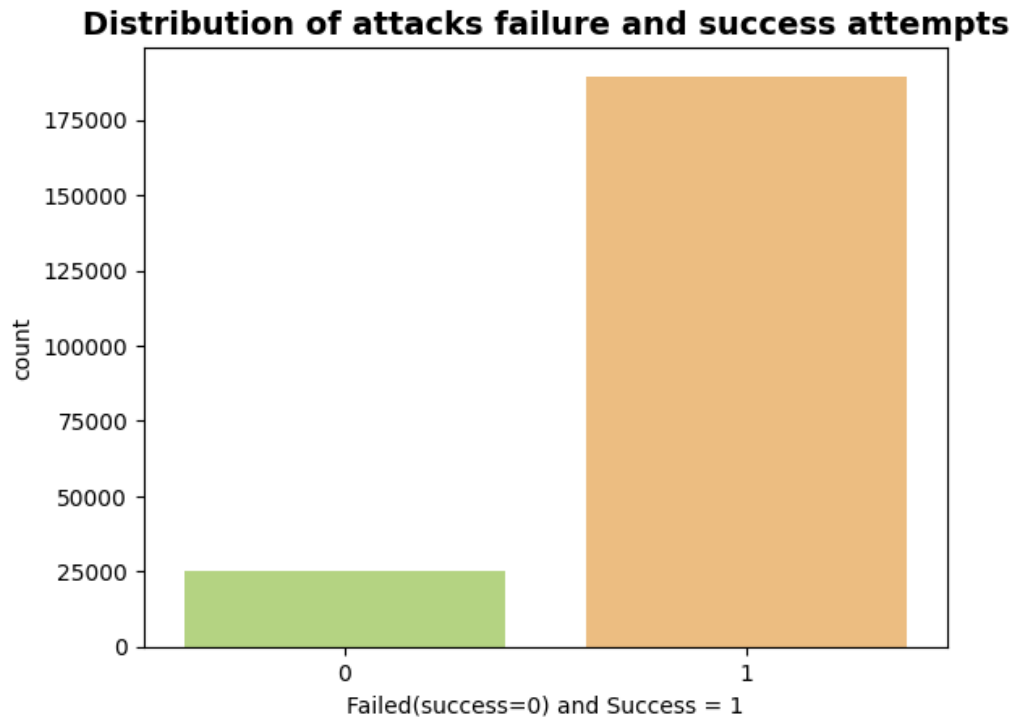
Hyperparameter tuning involves adjusting parameters like the learning rate, tree depth, and the number of boosting rounds to optimize the GBM's performance. Grid search or random search can be used for this purpose.

Experimental Results

We start off with cleaning the data. Upon analysis, we found null values and misnamed columns. We removed the null values and renamed the columns for ease of EDA/modeling.

```
[ ] Data_Reading_copy_new=Data_Reading_copy.rename(columns={'iyear':'Year','imonth':'Month','  
[ ] Data_Reading_copy_new=Data_Reading_copy_new[['Year','Day','Month','Date','Country','City'  
Data_Reading_copy_new.shape
```

We also noticed that the data is heavily imbalanced as depicted in the graph below



We use SMOTE to balance our dataset.

```
#Apply SMOTE and Split the data.
from imblearn.over_sampling import SMOTE

# Initialize SMOTE
sm = SMOTE(random_state=42)

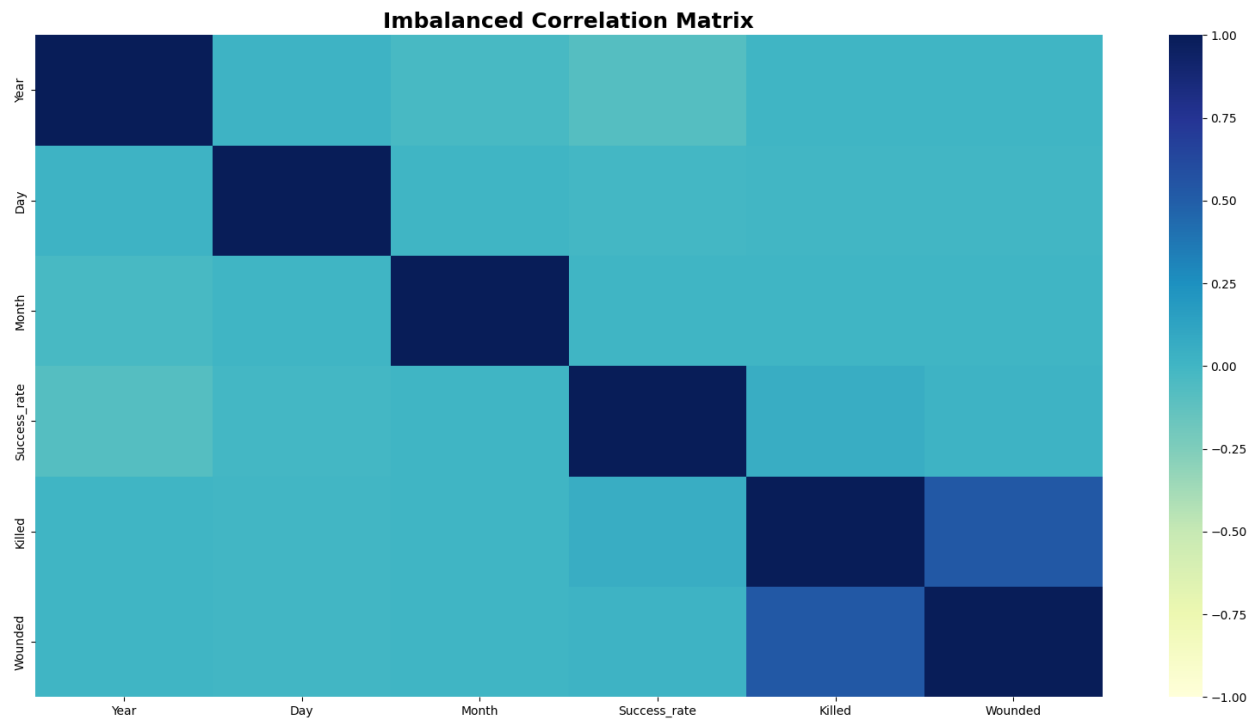
# Define features and target
features = Data_Reading10.drop(columns='success')
target = Data_Reading10['success']

# Apply SMOTE
X, y = sm.fit_resample(features, target)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
X_train.shape
```

Multicollinearity Handling:

Multicollinearity refers to the presence of high correlations between predictor variables, which can lead to instability and inaccurate coefficient estimates. To assess multicollinearity in the

dataset, we used two common approaches - utilizing a heatmap of variable correlations and calculating Variance Inflation Factor (VIF) scores.



	feature	VIF
0	Year	14.722601
1	Day	4.125666
2	Month	4.582517
3	Success_rate	8.055550
4	Killed	1.475686
5	Wounded	1.406061

Identifying and Dropping "Year":

- Identified high multicollinearity in the "Year" variable.
- Decision: Dropped the "Year" variable to address the multicollinearity issue.

Categorical Variable Encoding:

- Recognized that most variables in the dataset are categorical.
- Decision: Performed encoding to convert categorical variables into numerical format suitable for modeling.

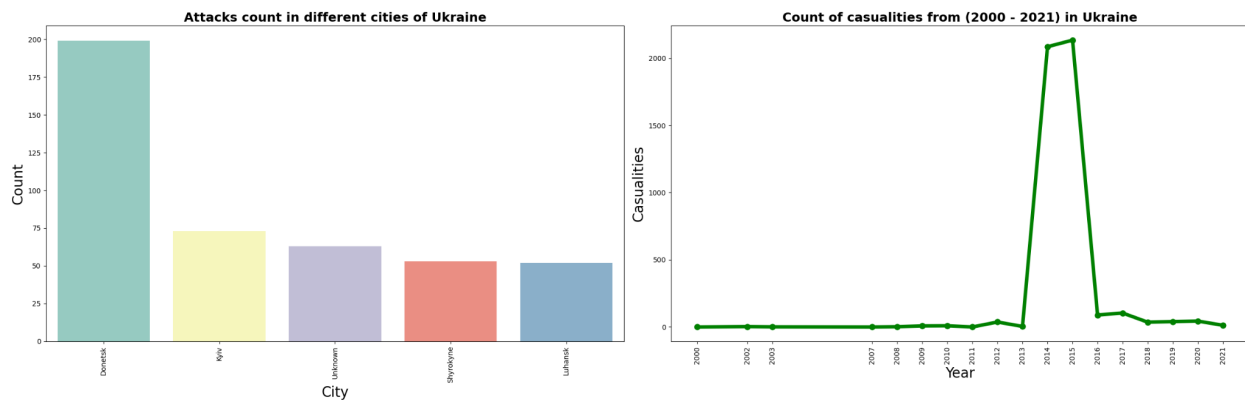
Handling Missing Values:

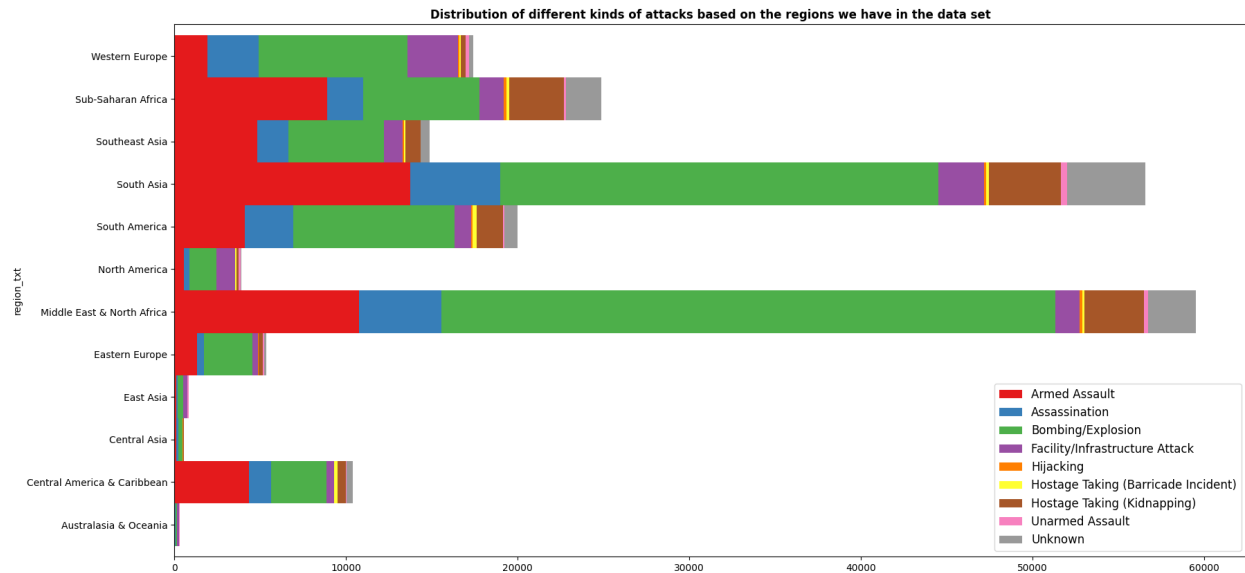
- Observed a substantial number of missing or null values in the dataset.
- Decision: Imputed missing values to address data incompleteness.
- Techniques may include mean imputation, median imputation, or more advanced methods based on the nature of the data.

Data Scaling:

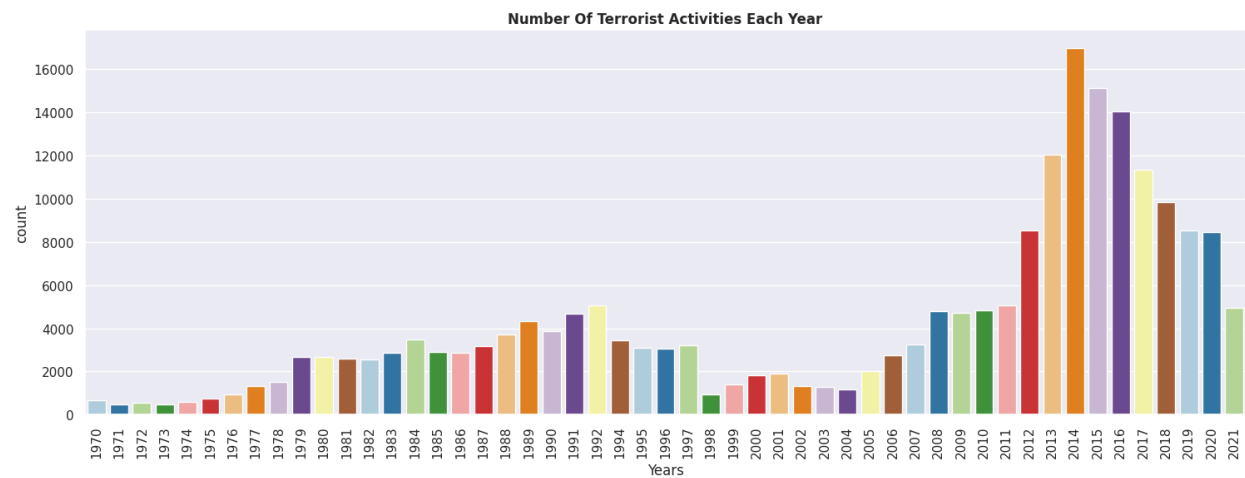
- Applied data scaling using StandardScaler from the sklearn package.
- Decision: Scaled the data to standardize the features, especially for the SVM algorithm.
- Purpose of Scaling: Helps algorithms that are sensitive to the scale of features, such as SVM. Ensures all features contribute equally to the model.

Visualizations and EDA

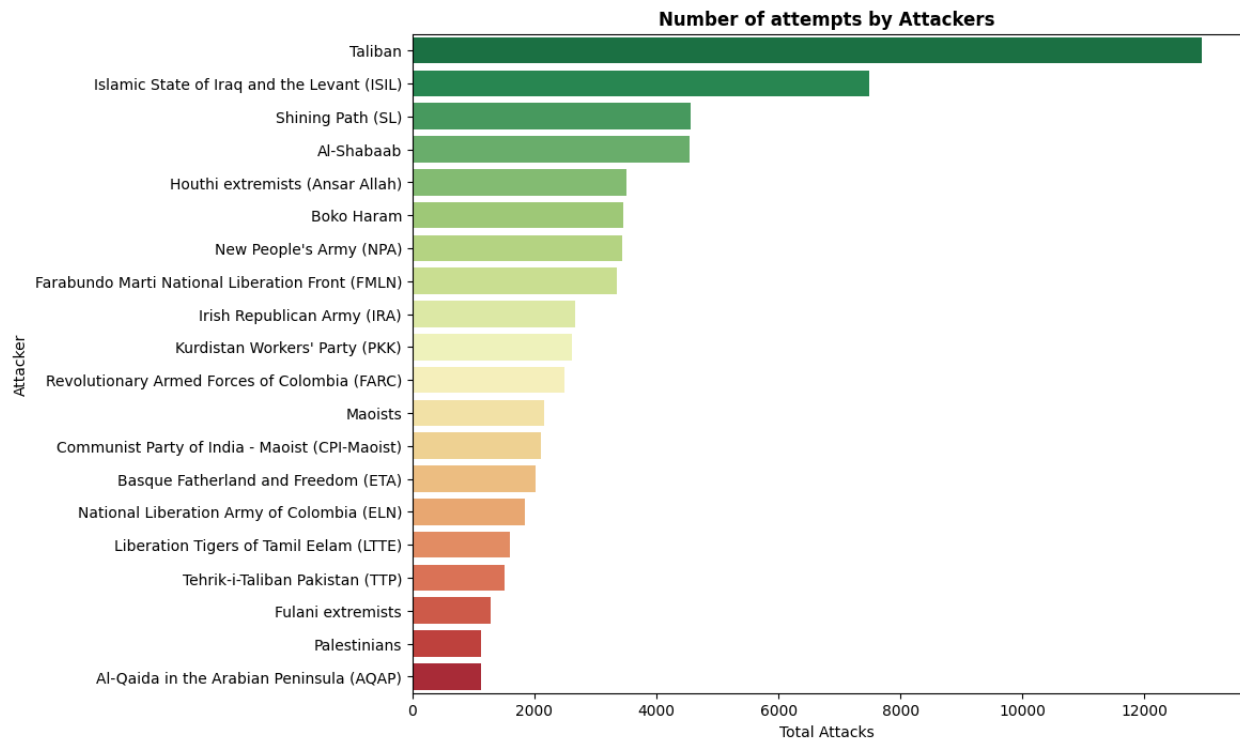




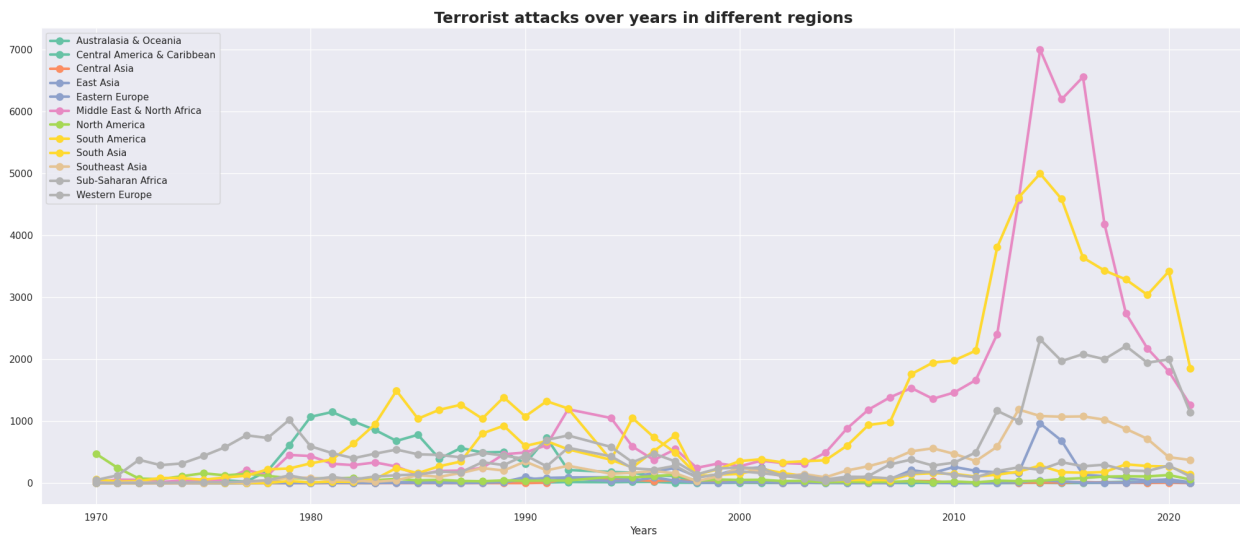
Bombing and Explosion seems to be a popular choice for terrorist groups followed by armed assassination



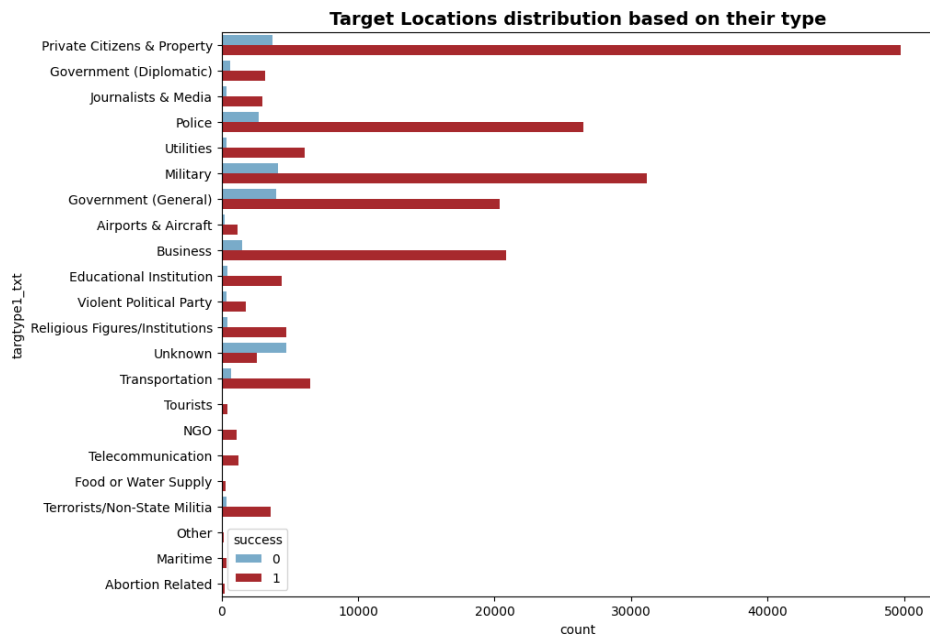
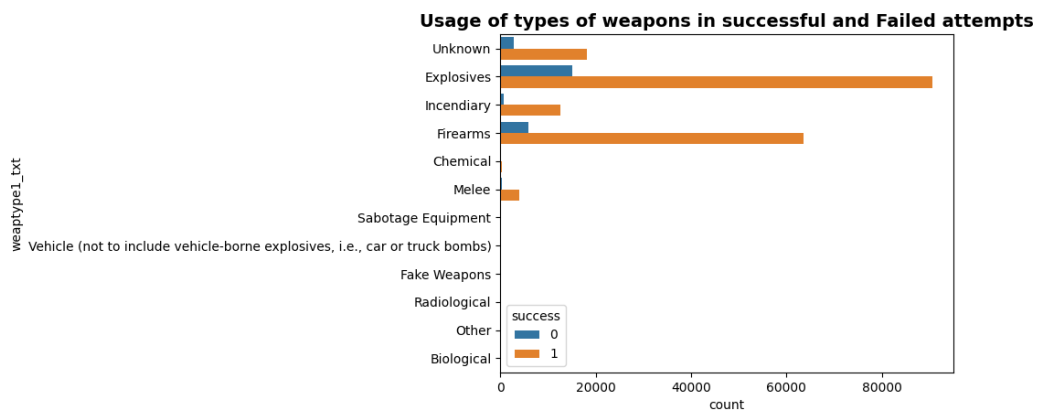
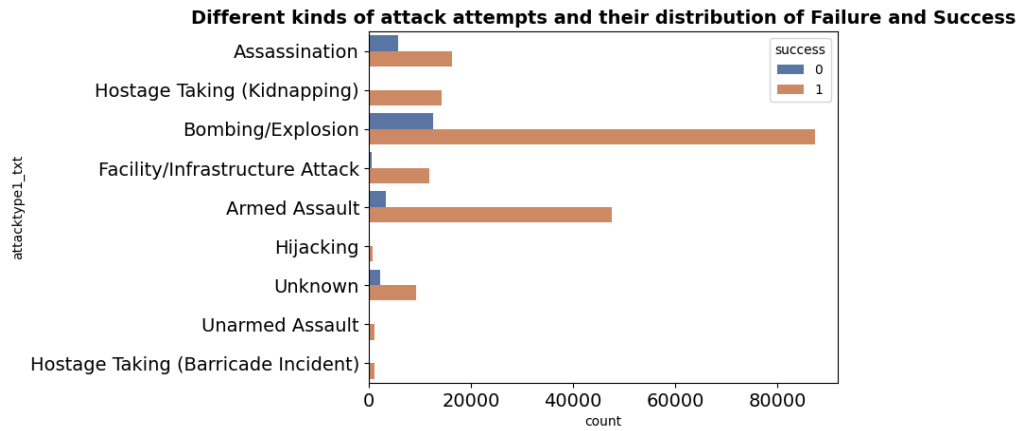
We can observe that the terrorist activities have been at it's peak in 2014 and there's been a steady decrease since then



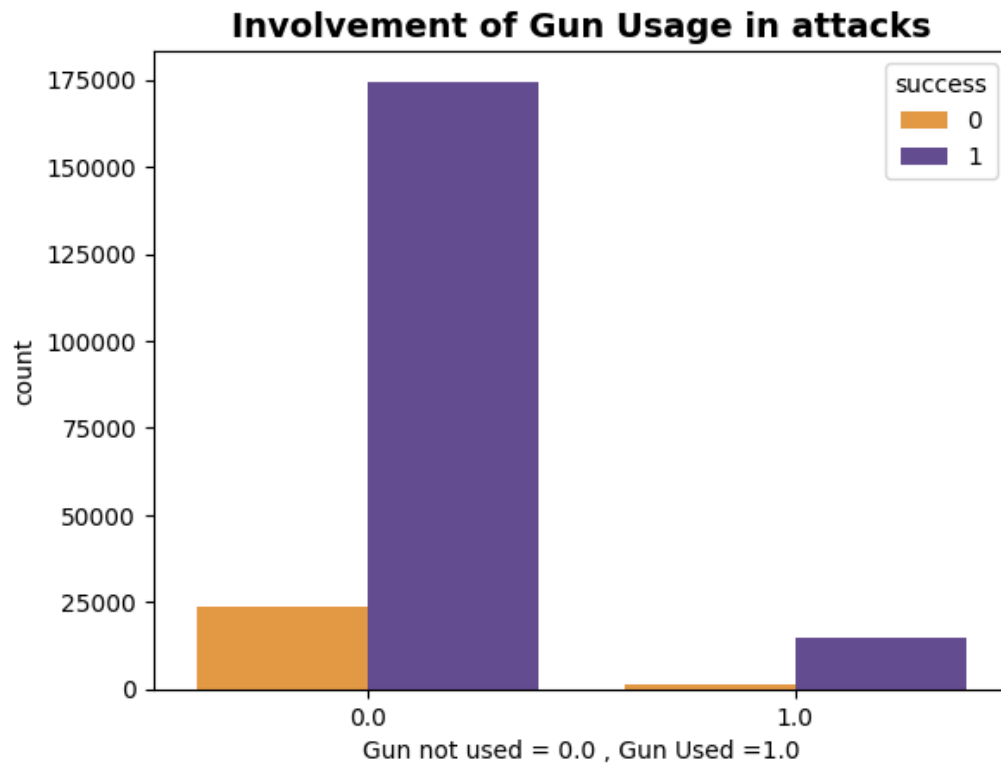
Here we see that the highest number of attempts have been made by the infamous Taliban and shows that more than 12000 attacks have been made by this terrorist group.



Here we see that Middle east and North Africa experienced the highest number of terrorist attacks. This has decreased considerably since close to 2020. South Asia comes next in this graph.




Plot shows the locations of terrorist attacks. It is usually seen that private properties or innocent citizens are the most targeted locations. Even more than the military base camps.



In the last plot we can see that, in 90% of terrorists activities gun crime is involved.

Model Output:

K Nearest Neighbours Model

```
 #KNN model
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report

# Create KNN classifier
knn = KNeighborsClassifier(n_neighbors=5)

# Train the model using the training sets
knn.fit(X_train, y_train)

# Predict the response for the test dataset
y_pred = knn.predict(X_test)

# Calculate and print accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Calculate and print the confusion matrix
cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", cm)

report = classification_report(y_test, y_pred)
print("Classification Report: \n", report)
```

Results

```
Accuracy: 0.8284205548549811
Confusion Matrix:
[[11389 1322]
 [ 3032 9633]]
Classification Report:
              precision    recall  f1-score   support

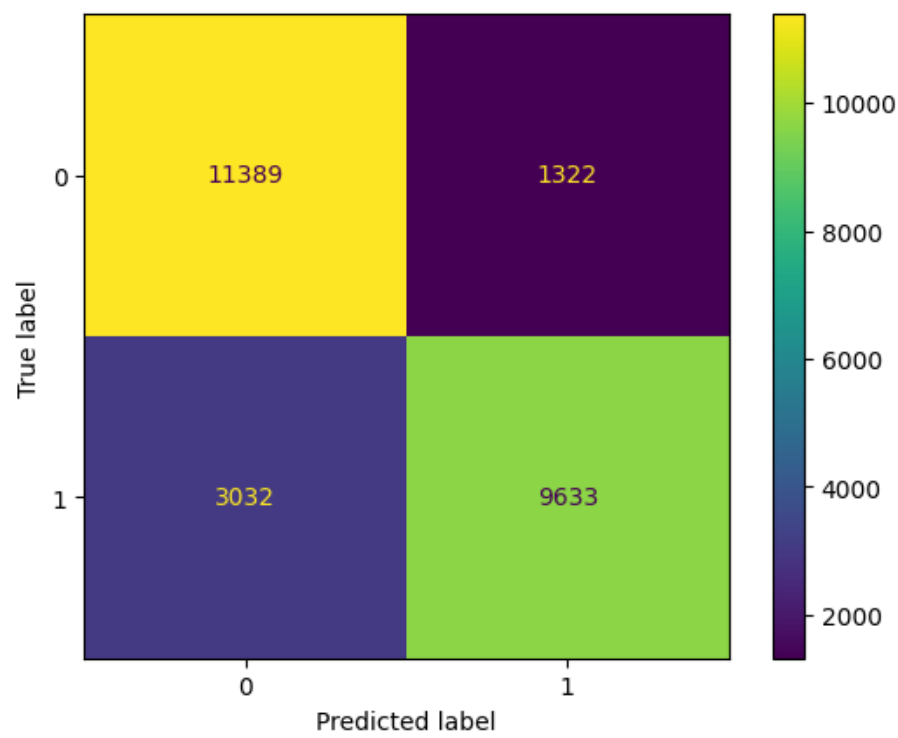
      0       0.79       0.90       0.84      12711
      1       0.88       0.76       0.82      12665

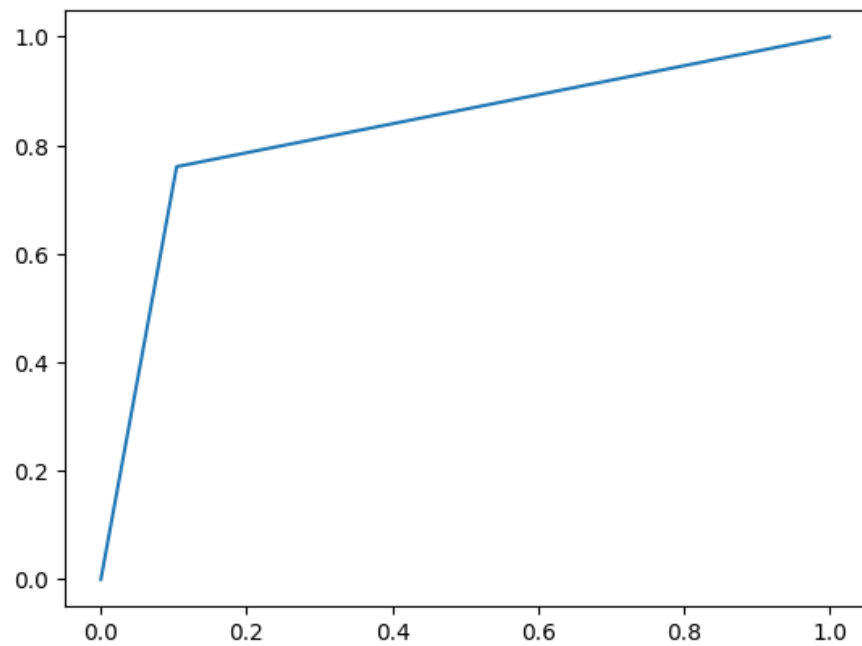
 accuracy              0.83      25376
 macro avg              0.83      25376
weighted avg              0.83      25376

] roc_auc_score(y_test, y_pred)

0.8282978366624206
```

Confusion Matrix





Strengths:

- The model demonstrates a commendable overall accuracy of 82.84%.
- High precision and recall values for both classes, indicating a balanced performance.

Considerations:

- Class 1 (positive class) has a slightly lower recall (76%), suggesting room for improvement in capturing all instances of terrorism.
- The confusion matrix highlights areas of correct and incorrect classifications.

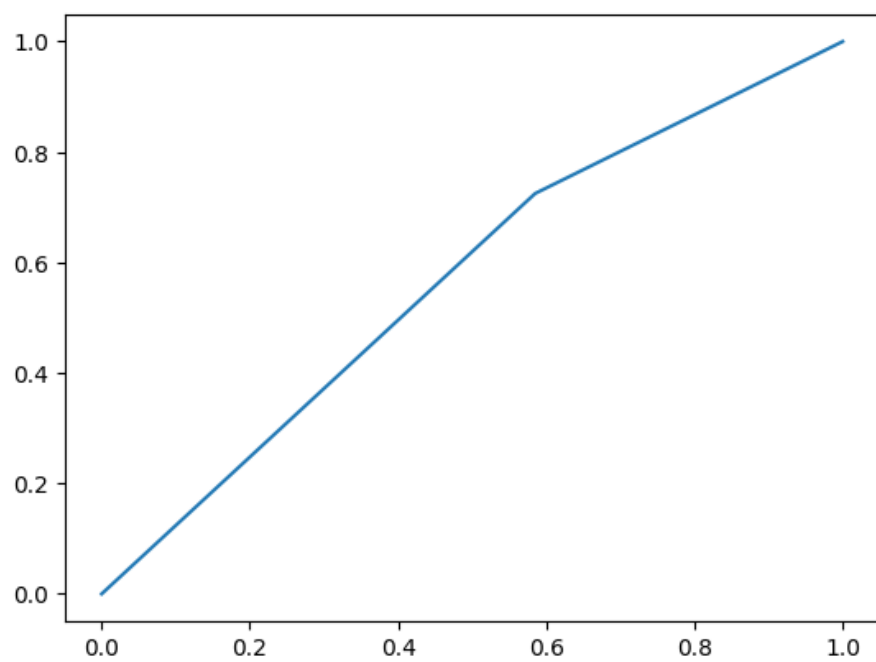
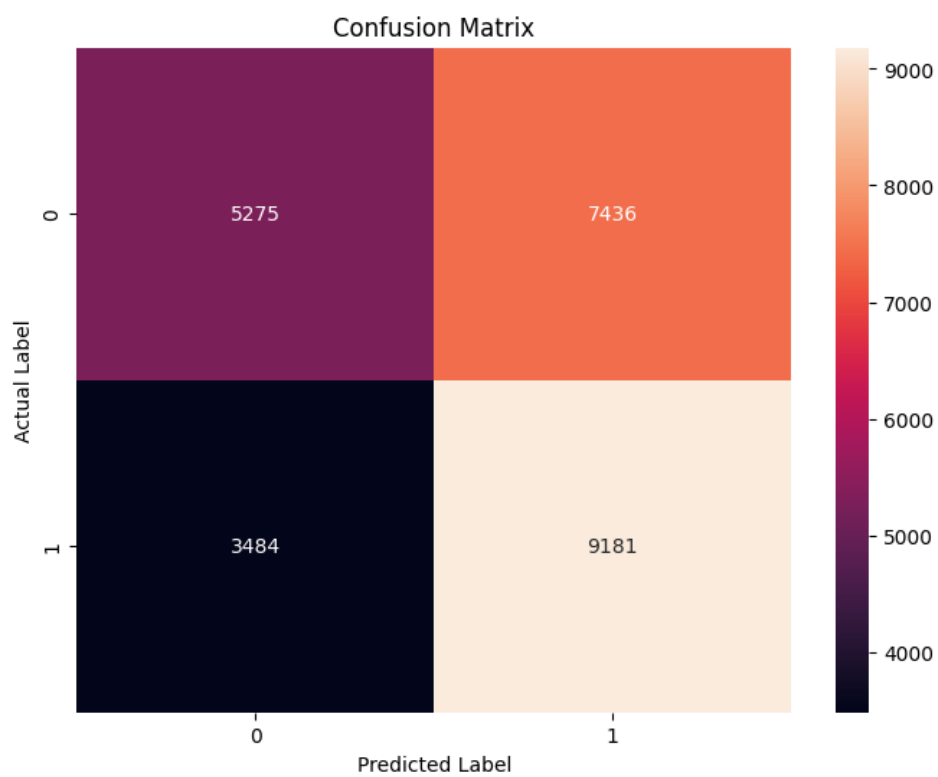
SVM model

```
Classification Report:
              precision    recall  f1-score   support

     0       0.60      0.41      0.49      12711
     1       0.55      0.72      0.63      12665

 accuracy          0.57      25376
 macro avg       0.58      0.57      0.56      25376
 weighted avg    0.58      0.57      0.56      25376

Accuracy: 0.569672131147541
0.5699530294208183
```



Strengths

The model achieved a balanced precision and recall for both classes.

Higher recall in Class 1 indicates a better ability to capture instances of terrorism.

Considerations:

Class 0 (negative class) has a lower recall (41%), suggesting challenges in identifying non-terrorism instances.

The overall accuracy is relatively modest.

The SVM model demonstrates a reasonable ability to distinguish between classes, with a notable focus on capturing instances of terrorism (Class 1). However, there is room for improvement, especially in correctly identifying non-terrorism instances (Class 0). Ongoing refinement and exploration of model parameters are essential to enhance predictive capabilities.

Applying PCA to SVM 1

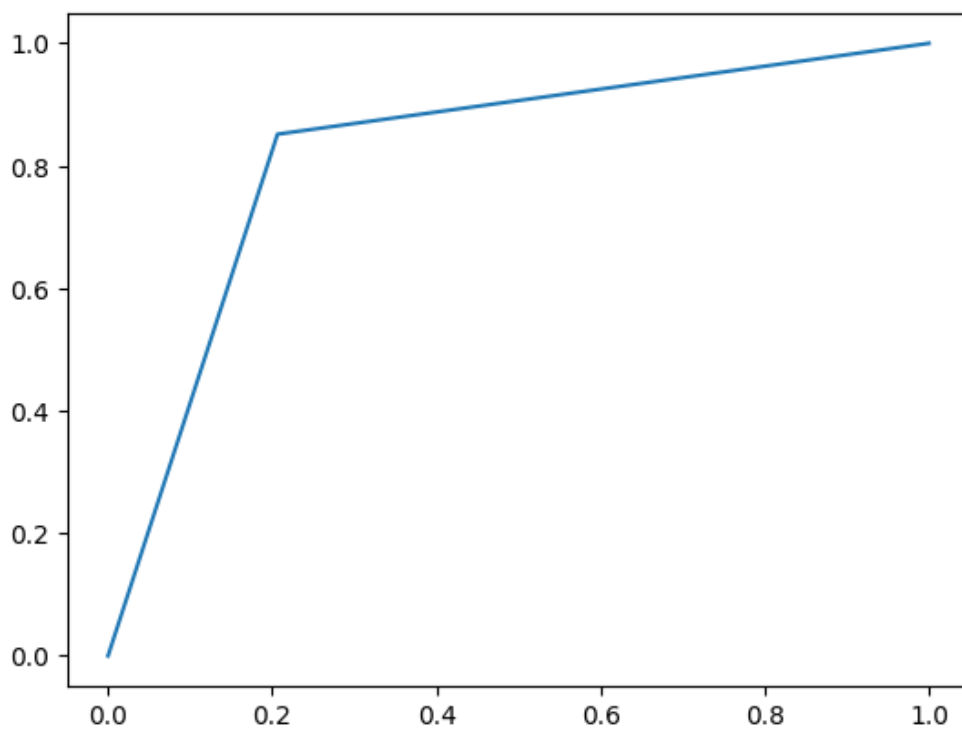
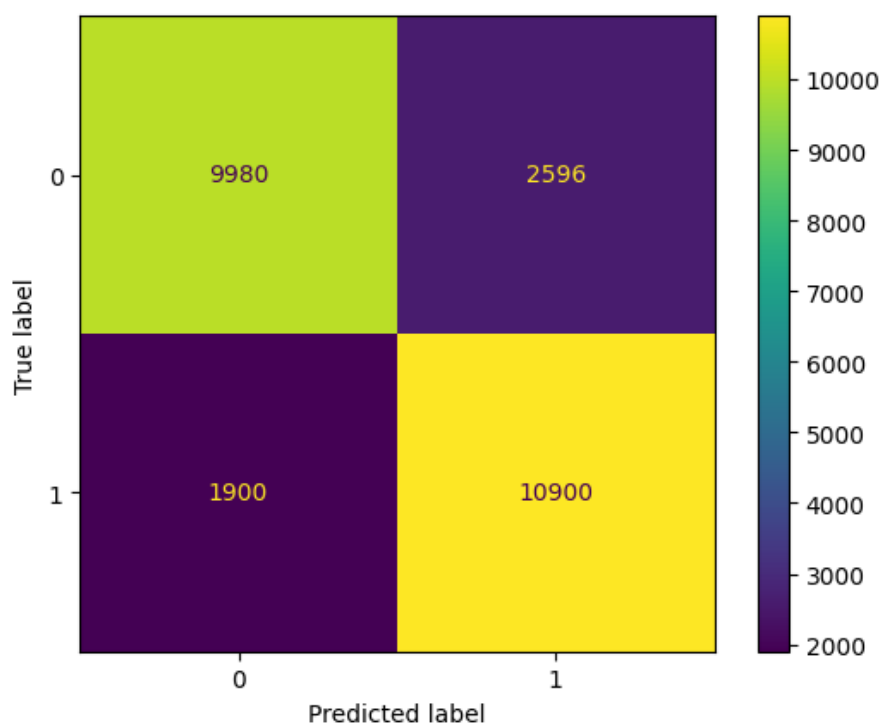
Confusion Matrix:

```
[[ 9980 2596]
 [ 1900 10900]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.84	0.79	0.82	12576
1	0.81	0.85	0.83	12800
accuracy			0.82	25376
macro avg	0.82	0.82	0.82	25376
weighted avg	0.82	0.82	0.82	25376

Accuracy: 0.8228247162673392
0.8225687818066156



- Accuracy: 82.28%
- The SVM model with PCA achieved an accuracy of 82.28%, indicating the proportion of correctly classified instances.

Strengths:

- Improved precision and recall for both classes compared to the SVM model without PCA.
- A balanced F1-score for both classes.

Considerations:

- The model demonstrates slightly better recall for Class 1 than Class 0.
- The overall accuracy is satisfactory.

The SVM model with PCA exhibits enhanced performance, achieving improved precision, recall, and F1-scores for both classes. The balanced results suggest that PCA has positively influenced the model's ability to distinguish between instances of terrorism and non-terrorism. Further refinements and investigations into the optimal number of principal components can contribute to the ongoing improvement of the model.

Applying PCA to SVM 2

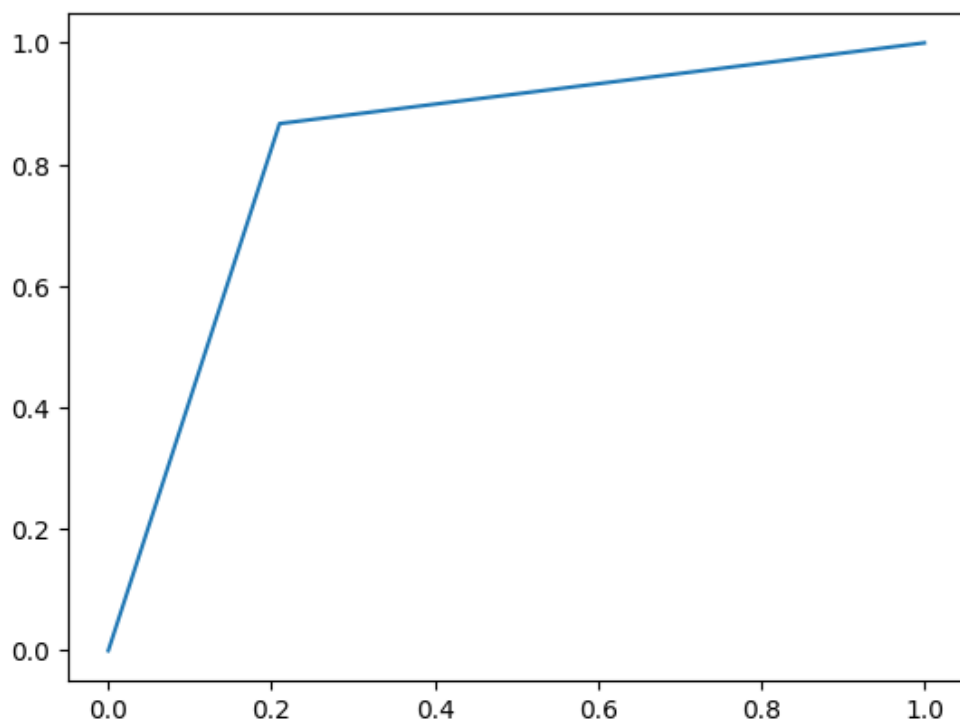
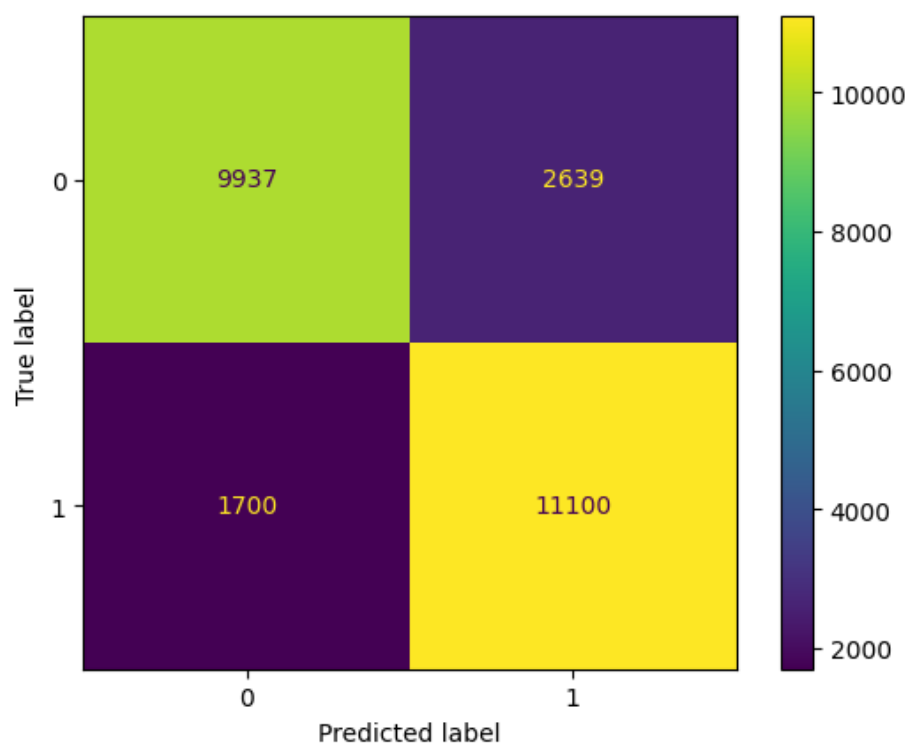
Confusion Matrix:

```
[[ 9937 2639]
 [ 1700 11100]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.85	0.79	0.82	12576
1	0.81	0.87	0.84	12800
accuracy			0.83	25376
macro avg	0.83	0.83	0.83	25376
weighted avg	0.83	0.83	0.83	25376

Accuracy: 0.8290116645649432



- Accuracy: 82.90%
- The SVM model 2 with PCA achieved an accuracy of 82.90%, indicating the proportion of correctly classified instances.

Strengths:

- Balanced precision and recall for both classes.
- Improved recall for Class 1 compared to SVM with PCA Model 1.

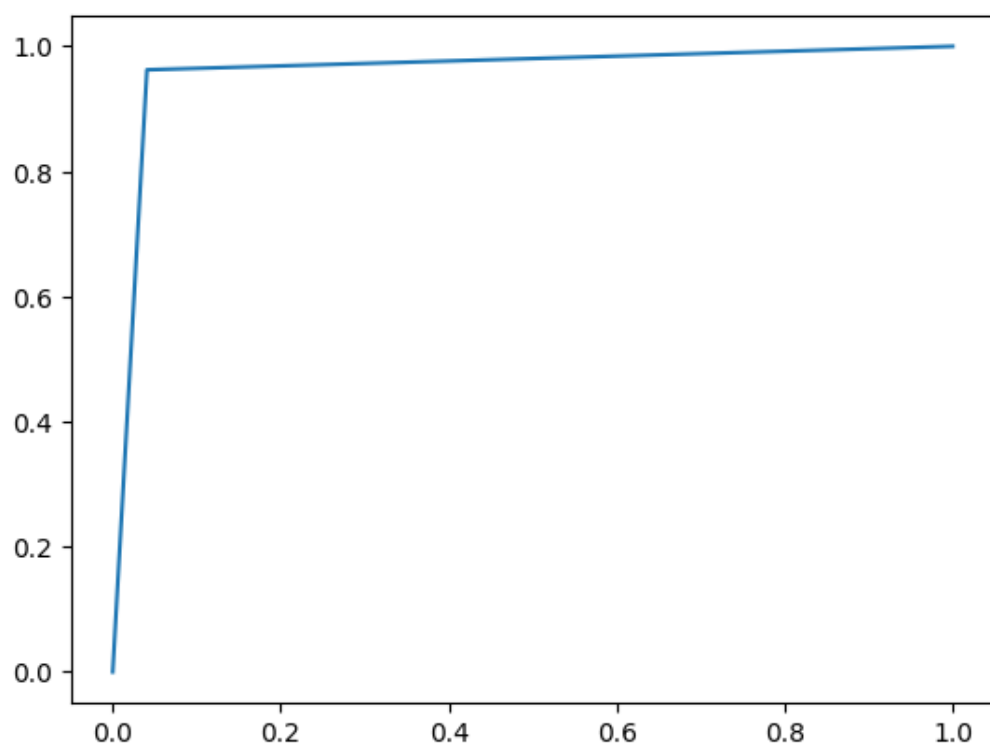
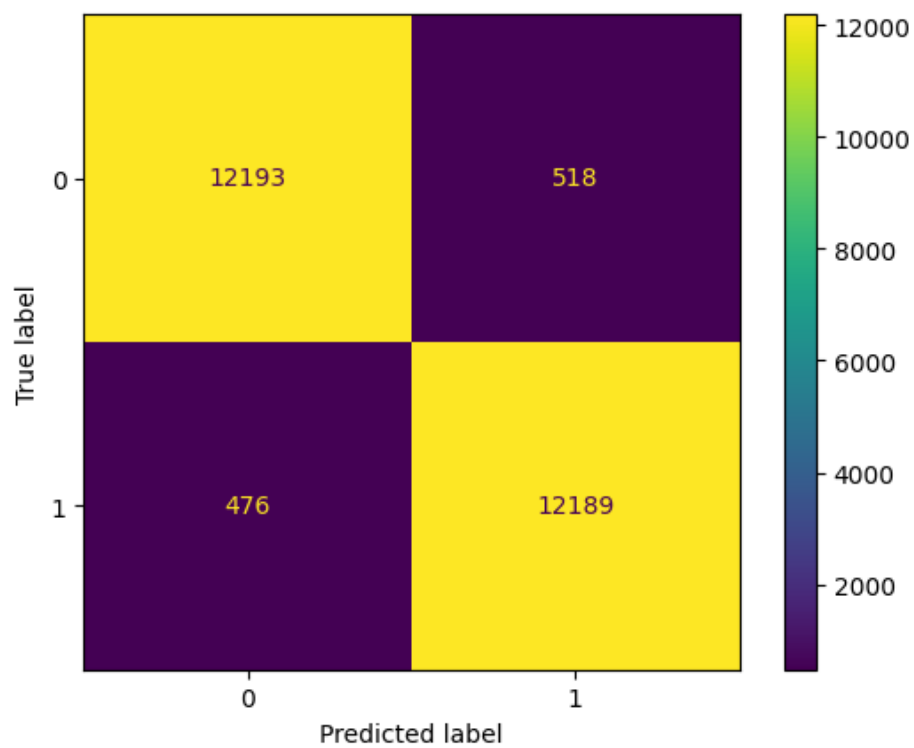
Considerations:

- Similar recall for Class 0 as in SVM with PCA Model 1.
- The overall accuracy is consistent with Model 1.

SVM with PCA Model 2 continues to demonstrate robust performance, achieving a well-balanced precision, recall, and F1-score for both classes. The slight variations in performance compared to Model 1 suggest ongoing refinements and exploration opportunities. Further investigations into the impact of different numbers of principal components and comparative analyses will contribute to the model's continuous improvement.

Gradient Boost Model

Accuracy: 0.9608291298865069					
Classification Report:					
	precision	recall	f1-score	support	
0	0.96	0.96	0.96	12711	
1	0.96	0.96	0.96	12665	
accuracy			0.96	25376	
macro avg	0.96	0.96	0.96	25376	
weighted avg	0.96	0.96	0.96	25376	
Confusion Matrix:					
[[12193 518]					
[476 12189]]					



Strengths:

Exceptional precision, recall, and F1-scores for both classes.

The model demonstrates high accuracy and well-balanced performance.

Considerations:

A very low number of false positives and false negatives, indicating the model's effectiveness.

The GBM stands out with exceptional accuracy, precision, recall, and F1-scores for both classes. Its ability to minimize false positives and false negatives showcases its robustness in predicting instances of terrorism. The model's high performance positions it as a strong candidate for deployment, pending further evaluation and consideration of resource implications. We used `scale_pos_weight` parameter by calculating the ratio of 0 values to 1 values.

Comparing models

Model	Accuracy	Precision (0)	Precision (1)	Recall (0)	Recall (1)	F1-Score (0)	F1-Score (1)
K-Nearest Neighbors	82.84%	79%	88%	90%	76%	84%	82%
Support Vector Machine	56.97%	60%	55%	41%	72%	49%	63%
SVM with PCA Model 1	82.28%	84%	81%	79%	85%	82%	83%
SVM with PCA Model 2	82.90%	85%	81%	79%	87%	82%	84%
Gradient Boosting	96.08%	96%	96%	96%	96%	96%	96%

GBM has the highest accuracy, precision, recall, and F-1 score out of all the models.

Conclusion:

Model Performance Overview:

- The Gradient Boosting Machine (GBM) model emerged as the standout performer, achieving an outstanding accuracy of 96.08%.
- K-Nearest Neighbors (KNN) showcased commendable accuracy and a well-balanced trade-off between precision and recall.
- Support Vector Machine (SVM) models, especially those with Principal Component Analysis (PCA), demonstrated improvement over the baseline SVM.

Model Selection and Deployment:

- Primary Recommendation: Considering the exceptional accuracy, precision, recall, and F1-scores, the GBM model is recommended for deployment in the terrorism prediction system.
- Reasoning: GBM not only achieved the highest accuracy but also demonstrated robustness in minimizing both false positives and false negatives, essential for effective terrorism prediction.

Key Findings and Considerations:

GBM Strengths:

- High accuracy and well-balanced precision and recall.
- Minimal false positives and false negatives, indicating a reliable predictive capability.

SVM with PCA Impact:

- SVM models with PCA showcased improvements over the baseline SVM, demonstrating the value of feature reduction techniques.