A
Minor Project Report on
"GLOBAL TERRORISM DATA ANALYSIS"

In partial fulfillment of requirements for the degree of

Bachelor of Technology (B. Tech.)

in

Computer Science and Engineering

Submitted by
Ms. Riya Khanna (170436)

Under the Guidance of

Dr. Suneet Gupta

Department of Computer Science and Engineering

SCHOOL OF ENGINEERING AND TECHNOLOGY

Mody University and Science and Technology
Lakshmangarh, Distt. Sikar-332311

December, 2020

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my mentor Dr. Suneet Gupta for providing his invaluable guidance, comments and suggestions throughout this project. His kindness and help have been the source of encouragement for me. I am grateful to him for the guidance, inspiration and constructive suggestions that helped me in completing this project. I would also like to extend my gratitude to Dr. A.Senthil, Professor and Assistant Dean Mody University, Laxmangarh, for giving me this opportunity to undertake this project

Riya Khanna

170436

CERTIFICATE

This is to certify that the minor project report entitled “Global Terrorism Data Analysis” submitted by Ms. Riya Khanna, as a partial fulfillment for the requirement of B. Tech. VII Semester examination of the School of Engineering and Technology, Mody University of Science and Technology, Lakshmangarh for the academic session 2019-2020 is an original project work carried out under the supervision and guidance of Dr. Suneet Gupta has undergone the requisite duration as prescribed by the institution for the project work.

PROJECT GUIDE:	**HEAD OF DEPARTMENT**
Approval Code: AUT_20_CSE_F11_02	**Signature:**
Name: Dr. Suneet Gupta	**Name:** Dr. A. Senthil
Date: 27th Dec, 2020	**Date:** 27th Dec, 2020

EXAMINER-I:	**EXAMINER-II**
Name: Mr. Hitesh Jangir	**Name:** Dr. Niranjan Lal
Dept: Computer Science and Engineering	**Dept:** Computer Science and Engineering

ABSTRACT

This project is made in order to determine if an event in the Global terrorism Database can be classified as exclusively terrorist or other forms of crime. Compared to most types of criminal violence, terrorism poses special challenges to a nation and exhausts all of its resources in prevention of it including the loss of life. While the human cost is devastating, the economic impact may be larger than most realize. Terrorism is one of the parameters that tourists check for, before visiting a country and hence if a Nation has more prevalent terrorism, chances are, despite its fascinating tourist attractions, it might end up in little to no Tourism. In response, there has been growing interest in researching about terrorism, their motives and most vulnerable target groups that are attacked. One thing that is infrequent is one common definition of Terrorism throughout the world. This is why the Global Terrorism Database has also included those events here which don’t confirm to global inclusion criteria for terrorism but are identified as terrorist events by the locals.

My aim in this project was to classify the events as terrorist or other forms of crime based on the Global inclusion parameters so that we can help the various intelligence agencies to drill down their study exclusive to Terrorism, avoiding any ambiguity that would come with the raw data(G.T.D).This would help them in formulation of the accurate responses to the same. Also, this analysis would help in understanding the dynamics, causes and consequences of terrorism around the world, analyzing patterns such as the frequency of terrorist attacks, the lethality of terrorist attacks, patterns of casualties etc. The extracted features from the data are fed to the machine learning classification methods to build a model. Feature selection pre-processing steps are used to enhance the performance and scalability of the classification methods.

Table of Contents

Chapter 1: Introduction

1. INTRODUCTION:

In order to do this project, I have made use of the Global Terrorism Database. The Global Terrorism Database (GTD) is an open-source database comprising of data and information about the terrorist attacks that have taken place around the globe from 1970 through 2017. The GTD contains regular data on domestic and international terrorist incidents that have taken place during this time period and today includes more than 200,000 attacks. In the database, information is available on the date and location of the incident, the weapons used which were used, the number of casualties that occurred, and the information about the perpetrator. The database is preserved by researchers at the National Consortium for the Study of Terrorism and Responses to Terrorism (START), which is headquartered at the University of Maryland.

1.1. Present System

In comparison to collection of other types of criminal violence data, terrorism data collection poses data collection challenges. Therefore, due to this issue there has been growing awareness in open source terrorist event databases. In the past such kind of databases have been limited to a single definition of terrorism and this is one of the most important problems with such kinds data bases.

According to the dictionary the definition of terrorism is “The defenseless or real utilization of unlawful power and cruelty by a non-state entertainer to accomplish a political, financial, strict, or social objective through dread, or terror."

But in some of the cases, it may occur that there may be some doubt whether an attack fulfills all the conditions to come under the category of terrorism. In such cases, where there is a strong chance, but not certainty, that an incident comes under the act of terrorism, the incident is included in GTD under a feature – “Doubt Terrorism Proper”.

1.2. Proposed System

This project is made in order to determine if an event in the Global terrorism Database can be classified as exclusively terrorist or other forms of crime. The need for such kind of analysis is required as for many years identifying the causes of terrorism have been a goal of many researchers. And over the years, a huge increase in terrorist activities has been observed. But, the motives behind such terrorist activities have remained unknown. Hence, the ultimate goal of this project is to find some causes of terrorism in order to predict and/or reduce future incidents.

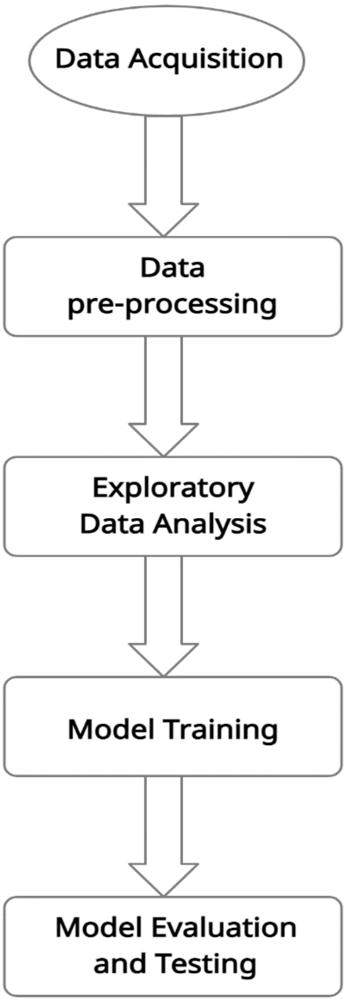
This analysis would help in understanding the dynamics, causes and consequences of terrorism around the world, analyzing patterns such as the frequency of terrorist attacks, the lethality of terrorist attacks, patterns of casualties etc. Hence, this can improve the understanding of terrorist violence so that it can be more readily studied and defeated.

The findings from this project can be used by the Center for terrorism and Intelligence, (United States), to research and exclusively work on events pertaining to terrorism and find repeating patterns across the globe. It will also help in classifying the event into other types of crime if it is not exclusively terrorist.

This work can be used by curious civilians, security related policy-makers, international organizations hosting worldwide events, foreign investors and academic researchers for the purpose of understanding terrorism and its nature.

2. SYSTEM DESIGN

2.1. System Flowchart



2.2. Data Dictionary

The dataset has 135 columns and a few important columns are mentioned below:

- **eventid-** this is the Unique ID of the event. (Numeric Variable)
- **iyear**-year of the incident (Numeric Variable)
- **imonth**-month of the incident (Numeric Variable)

- **iday** – Day of the incident (Numeric Variable)
- **extended-** Extended Incident (Categorical Variable)
- **summary** -Incident Summary (Text Variable)
- **crit1, crit2, crit3**-Inclusion Criteria (Categorical Variables)
- **multiple** – have been a part of multiple incident (Categorical Variable)
- **related** - related incidents (Text Variable)
- **country, country_txt**- country (Categorical Variable)
- **region, region_txt-** region (Categorical Variable)
- **city** - city (Text Variable)
- **latitude** – latitude (Numeric Variable)
- **longitude** - longitude (Numeric Variable)
- **specificity** -Geo-Coding Specificity (Categorical Variable)
- **attacktype1, attacktype1_txt** – Attack Type (Categorical Variable)
- **success** -Successful Attack (Categorical Variable)
- **suicide** -Suicide Attack (Categorical Variable)
- **weaptype1, weaptype1_txt** -Weapon Type (Categorical Variable)
- **targtype1, targtype1_txt** -Victim Type (Categorical Variable)
- **corp1**: Name of the Entity (Text Variable)
- **natlty1; natlty1_txt** -Nationality of the Victim (Categorical Variable)
- **individual** -Unaffiliated Individuals (Categorical Variable)
- **gname** -Perpetrator Group Name (Text Variable)
- **nperps**- Number of the Perpetrators (Numeric Variable)
- **claimed** -Claim of Responsibility (Categorical Variable)
- **motive** - Motive (Text Variable)
- **Nkill** - Total Fatalities - (Numeric Variable)
- **nkillter** - perpetrator Fatalities (Numeric Variable)
- **nwound** -Total Injured (Numeric Variable)
- **nwoundte** - Perpetrators Injured (Numeric Variable)
- **property** - property damage (Categorical Variable)
- **propvalue**- Value of the Property Damage (in usd) (Numeric Variable)
- **nhostkid**- Total No. Of Hostages / Victim Kidnaps (- Numeric Variable)

- **ransomamt** -Total Ransom Amount Demanded (Numeric Variable)
- **ransompaid** -Total Ransom Amount Paid (Numeric Variable)
- **INT_IDEO** – Attack of an International Ideology (Categorical Variable)

Target Variable:

doubtterr - Doubt Terrorism Proper (Categorical Variable)

Information: Multiclass classification (0: Terrorism,1: doubtful, -9: unknown)

3. HARWARE AND SOFTWARE DETAILS

3.1. Software Details

Google Colab: Google Colaboratory is an item from Google Research. Colab permits anyone to compose and execute self-assertive python code through the program, and is particularly appropriate in fileds like Artificial Intelligence, machine learning and Neural Networks. All the more actually, Colab is a facilitated Jupyter scratch pad administration that requires no arrangement to utilize, while giving free admittance to figuring assets including GPUs.

Google Colaboratory is a cloud administration that can be utilized for nothing of cost, given by Google. It bolsters free GPU and depends on Google Jupyter Notebooks environment. It gives a platform to anybody to grow profound learning applications utilizing usually utilized libraries, for example, PyTorch, TensorFlow and Keras. It gives a route to your machine to not convey the heap of weighty exercise of your ML activities. It is one of the exceptionally famous foundation of the sort.

Chapter 4: Implementation and Work Details

4. IMPLEMENTATION WORK DETAILS

4.1. Real Life Applications

The need for such kind of analysis is required as for many years identifying the causes of terrorism have been a goal of many researchers. And over the years, a huge increase in terrorist activities has been observed. But, the motives behind such terrorist activities have remained unknown. Hence, the ultimate goal of this project is to find some causes of terrorism in order to predict and/or reduce future incidents.

This analysis would help in understanding the dynamics, causes and consequences of terrorism around the world, analyzing patterns such as the frequency of terrorist attacks, the lethality of terrorist attacks, patterns of casualties etc. Hence, this can improve the understanding of terrorist violence so that it can be more readily studied and defeated. This project would be answering several questions which need to be answered in order to prevent and predict such similar future incidents. Some of the questions that this analysis is aiming to answer are: Trend of global Terrorism from year to year, top countries having the highest Terrorism rate, identifying the terrorist groups causing the most acts, most common targets of the terrorist groups, most affected countries, types of weapons used, different types of attacks, terrorism trend particularly in India, and many more such similar questions.

Also, this project will determine if an event in the Global terrorism Database can be classified as exclusively terrorist or other forms of crime. Compared to most types of criminal violence, terrorism poses special challenges to a nation and exhausts all of its resources in prevention of it including the loss of life. The findings from this project can be used by the Center for terrorism and Intelligence, (United States), to research and exclusively work on events pertaining to terrorism and find repeating patterns across the globe.

4.2. Data Implementation and Program Execution

As any other Machine Learning project, the very first step of the process is understanding the problem. This initial step is the place where the goal is characterized. An understanding of how the machine learning system's solution will ultimately be used is important. This progression is likewise where tantamount situations and current workarounds to a given issue are examined, as well as assumptions being contemplated, and the degree of need for human expertise determined. Moving further on, the data was collected and pre-processed. After data cleaning, the data was visualized in order to get insights about how Terrorism has affected countries around the globe and how it has changed over time. Moreover, many questions are answered such as the numbers of worldwide killings, the types of attacks, motives of the attacks, terrorist groups involved, the different weapons they've been using and the countries they are targeting, trend of global Terrorism from year to year, top countries having the highest Terrorism rate, identifying the terrorist groups causing the most acts, most common targets of the terrorist groups, most affected countries, types of weapons used, different types of attacks and so on.

After data processing and visualization various classification algorithms are used to classify the Attacks based on set of independent variables like : Number of fatalities (terrorist an d targets), No of Wounded (terrorist and targets)Terrorists captures, longitude, latitude, extended events, three inclusion criteria, Multiple incidents, country region, city, specificity, attack type, success s of attack, Suicide attacks, Weapons used, Target type, Nationality of target groups, Individual or group attack, Terrorist Group Name, Claimed attacks, Property damage. Precision, Recall and Specificity, are three major performance metrics used to evaluate various models. Model Evaluation was done on all the models and the best suited model was decided based on performance.

Chapter 5: Source Code

5. SOURCE CODE

5.1 Using the Code

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import math
import warnings
warnings.filterwarnings("ignore")
df= pd.read_csv("data.csv",encoding='ISO-8859-1',low_memory=False)
#Specifies the percent of data that is blank for each column
len_df = len(df)
percent_null = []
for item in df.columns:
    count_null = df[item].isnull().sum()
    percent_null.append(count_null/len_df*100)
percent_column_null=pd.DataFrame({'column_name':df.columns,
'percent_null(%)':percent_null})
percent_column_null
#Drop Missing Value column> 50%
```

```
column_to_drop = []

for i in range(len(percent_column_null)):

    if percent_column_null['percent_null(%)'][i] > 50:

        column_to_drop.append(percent_column_null['column_name'][i])

df2 = df.drop(column_to_drop, axis=1)

len(df2.columns)

df2.info()

# Make a list of column names used for analysis

column_to_use = [ 'latitude', 'longitude', 'nperpcap', 'nkill', 'nkillter',

       'nwound', 'nwoundte', 'eventid', 'iyear', 'imonth', 'iday', 'extended',

       'crit1', 'crit2', 'crit3', 'multiple', 'country', 'country_txt',

       'region', 'region_txt', 'city', 'specificity', 'attacktype1',

       'attacktype1_txt', 'success', 'suicide', 'weaptype1', 'weaptype1_txt',

       'targtype1', 'targtype1_txt', 'natlty1', 'natlty1_txt', 'individual',

       'gname', 'claimed', 'property', 'doubtterr','provstate']

df2 = df[column_to_use]

df2.info()

df2.nunique()

#Create df_test_city which is City, Lat, and Long data that is not null

df_test_city = df2[(df2['city'].notnull()) & (df2['latitude'].notnull()) &
(df2['longitude'].notnull())]

#Create df_predict_city which is city data that is empty, but Lat and Long are not null
```

```
df_predict_city = df2[(df2['city'].isnull()) & (df2['latitude'].notnull()) & (df2['longitude'].notnull())]

len(df_predict_city)

def fill_city(df):

    lat_diff = np.abs(df['latitude'] - df_test_city['latitude'])

    long_diff = np.abs(df['longitude'] - df_test_city['longitude'])

    diff = lat_diff+long_diff

    index_min = np.argmin(diff,axis=0,out=None)

    return df_test_city['city'].loc[index_min]

idx_city_change = df_predict_city.index

df_test_city=df_test_city.reset_index(drop=True)

# Fill in the empty city data using apply

df2.loc[idx_city_change, 'city'] = df_predict_city.apply(fill_city, axis=1)

df2.isnull().sum()

df2 = df2.dropna(subset=['city'])

df_test_province = df2[(df2['provstate'].notnull()) & (df2['latitude'].notnull()) & (df2['longitude'].notnull())]

df_predict_province = df2[(df2['provstate'].isnull()) & (df2['latitude'].notnull()) & (df2['longitude'].notnull())]

len(df_predict_province)

def fill_province(df):

    lat_diff = np.abs(df['latitude'] - df_test_province['latitude'])

    long_diff = np.abs(df['longitude'] - df_test_province['longitude'])
    diff = lat_diff+long_diff

    index_min = np.argmin(diff)

    return df_test_province['provstate'].loc[index_min]

idx_prov_change = df_predict_province.index

df_test_province=df_test_province.reset_index(drop=True)

df2.loc[idx_prov_change, 'provstate'] = df_predict_province.apply(fill_province, axis=1)

df2 = df2.dropna(subset=['provstate'])

df2[df2.latitude.isnull()]['city'].value_counts()

df2[df2.latitude.isnull()]['country'].value_counts()

mean_lat_country = df2.groupby('country').mean()['latitude']

mean_long_country = df2.groupby('country').mean()['longitude']

# Create a function to fill in the blank latitude and longitude with the mean of the latitude or longitude of the country

def func_lat(df):

    if pd.isnull(df.latitude):

        return mean_lat_country[df.country]

    else:

        return df.latitude


def func_long(df):

    if pd.isnull(df.longitude):

        return mean_long_country[df.country]

    else:
        return df.longitude

df2['latitude'] = df2.apply(func_lat, axis=1)

df2['longitude'] = df2.apply(func_long, axis=1)

# Delete Lat and Long data that is still empty

df2.dropna(subset=['latitude', 'longitude'], inplace=True)

df_no_na = df2.dropna()

cat_column = []

for i in range(len(df2.columns)):

    if df2.nunique()[i] < 25:

        cat_column.append(df2.nunique().index[i])

        df2[df2.nunique().index[i]] = df2[df2.nunique().index[i]].astype('category')

cat_column

df.head(5)

df.shape

df.columns

import pandas as pd

df= pd.read_csv("data.csv",encoding='ISO-8859-1',low_memory=False)

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import math
import warnings
warnings.filterwarnings("ignore")
df.head(5)
df.shape
df.columns
plt.figure(figsize=(10,7))
corr_back = df.corr()
mask = np.zeros_like(corr_back, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True
sns.heatmap(corr_back, mask=mask, center=0, square=True, linewidths=.5,cmap='rainbow')
plt.show()
plt.subplots(figsize=(6,4))
sns.countplot(df['doubtterr'],palette='magma')
df['doubtterr'].value_counts(normalize=True)*100
plt.subplots(figsize=(15,6))
sns.countplot('iyear',data=df,palette='RdYlGn_r',edgecolor=sns.color_palette('dark',7))
plt.xticks(rotation=90)
plt.title('Number Of Terrorist Activities Each Year')
plt.show()
"""### **2. Attacking Methods used by Terrorists**"""
plt.subplots(figsize=(15,6))
sns.countplot('attacktype1_txt',data=df,palette='inferno',order=df['attacktype1_txt'].value_counts().index)
```

```
plt.xticks(rotation=90)

plt.title('Attacking Methods used by Terrorists')

plt.show()

plt.subplots(figsize=(15,6))

sns.countplot(df['targtype1_txt'],palette='inferno',order=df['targtype1_txt'].value_counts().index)

plt.xticks(rotation=90)

plt.title('Favorite Targets')

plt.show()

plt.subplots(figsize=(15,6))

sns.countplot('region_txt',data=df,palette='RdYlGn',edgecolor=sns.color_palette('dark',7),order=df['region_txt'].value_counts().index)

plt.xticks(rotation=90)

plt.title('Number Of Terrorist Activities By Region')

plt.show()

terror_region=pd.crosstab(df.iyear,df.region_txt)

terror_region.plot(color=sns.color_palette('Set2',12))

fig=plt.gcf()

fig.set_size_inches(18,6)

plt.title('Yearly Number Of Terrorist Activities By Region')

plt.show()

pd.crosstab(df.region_txt,df.attacktype1_txt).plot.barh(stacked=True,width=1,color=sns.color_palette('RdYlGn',9))
```

```
fig=plt.gcf()

fig.set_size_inches(12,8)

plt.title('Attacks in every Region')

plt.show()

plt.subplots(figsize=(18,6))

sns.barplot(df['country_txt'].value_counts()[:15].index,df['country_txt'].value_counts()[:15].values,palette='inferno')

plt.title('Top Affected Countries')

plt.show()

sns.barplot(df['gname'].value_counts()[1:15].values,df['gname'].value_counts()[1:15].index, palette=('inferno'))

plt.xticks(rotation=90)

fig=plt.gcf()

fig.set_size_inches(10,8)

plt.title('Terrorist Groups with Highest Terror Attacks')

plt.show()

df.info()

df.columns

df1=df.dropna(axis=0)

df1=df1.drop(['Unnamed: 0','eventid','iyear','imonth','iday','country_txt','region_txt','attacktype1_txt','weaptype1_txt','targtype1_txt','natlty1_txt'],axis=1)

# Labelencoding the 'city' and 'gname' variables

from sklearn import preprocessing
```

```
label_encoder = preprocessing.LabelEncoder()
df1['city']= label_encoder.fit_transform(df1['city'])
df1['gname']= label_encoder.fit_transform(df1['gname'])
pd.set_option('max_columns', None)
df1.head()
import sklearn
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, roc_auc_score, confusion_matrix
X=df1.drop(['doubtterr'],axis=1)
Y=df1['doubtterr']
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
X=sc.fit_transform(X)
X_train,X_test,Y_train,Y_test=train_test_split(X,Y, test_size=0.3, random_state=2)
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
model.fit(X_train,Y_train)
Y_pred=model.predict(X_test)
sklearn.metrics.accuracy_score(Y_test,Y_pred)
# print classification report
print(classification_report(Y_test,Y_pred))
```

```
cm=confusion_matrix(Y_test,Y_pred)

conf_matrix=pd.DataFrame(data=cm,columns=['Predicted:0','Predicted:1'],index=['Actual:0
','Actual:1'])

plt.figure(figsize = (8,5))

sns.heatmap(conf_matrix, annot=True,fmt='d',cmap="YlGnBu")

b, t = plt.ylim()

b += 0.5

t -= 0.5

plt.ylim(b, t)

plt.show()

from sklearn.model_selection import KFold

from sklearn.model_selection import cross_val_score

name='Logistic Regression'

kfold = KFold(shuffle=True,n_splits=5,random_state=1)

cv_results = cross_val_score(model, X_train,Y_train,cv=kfold, scoring='f1_weighted')

print("%s: %f (%f)" % (name, np.mean(cv_results),np.var(cv_results,ddof=1)))

print('cross validation scores: ',cv_results)

print('Bias error: ',np.mean((1-cv_results)*100))

print('variance error: ',np.var((1-cv_results)*100,ddof=1))

from sklearn.tree import DecisionTreeClassifier

model=DecisionTreeClassifier(criterion='entropy')

model.fit(X_train,Y_train)

Y_pred=model.predict(X_test)
```

```
sklearn.metrics.accuracy_score(Y_test,Y_pred)

print(classification_report(Y_test,Y_pred))

cm=confusion_matrix(Y_test,Y_pred)

conf_matrix=pd.DataFrame(data=cm,columns=['Predicted:0','Predicted:1'],index=['Actual:0','Actual:1'])

plt.figure(figsize = (8,5))

sns.heatmap(conf_matrix, annot=True,fmt='d',cmap="YlGnBu")

b, t = plt.ylim()

b += 0.5

t -= 0.5

plt.ylim(b, t)

plt.show()

name='Decision Tree'

kfold = KFold(shuffle=True,n_splits=5,random_state=1)

cv_results = cross_val_score(model, X_train,Y_train,cv=kfold, scoring='f1_weighted')

print("%s: %f (%f)" % (name, np.mean(cv_results),np.var(cv_results,ddof=1)))

print('cross validation scores: ',cv_results)

print('Bias error: ',np.mean((1-cv_results)*100))

print('variance error: ',np.var((1-cv_results)*100,ddof=1))

from sklearn.ensemble import RandomForestClassifier

model= RandomForestClassifier(n_estimators = 100)

model.fit(X_train,Y_train)

Y_pred=model.predict(X_test)
```

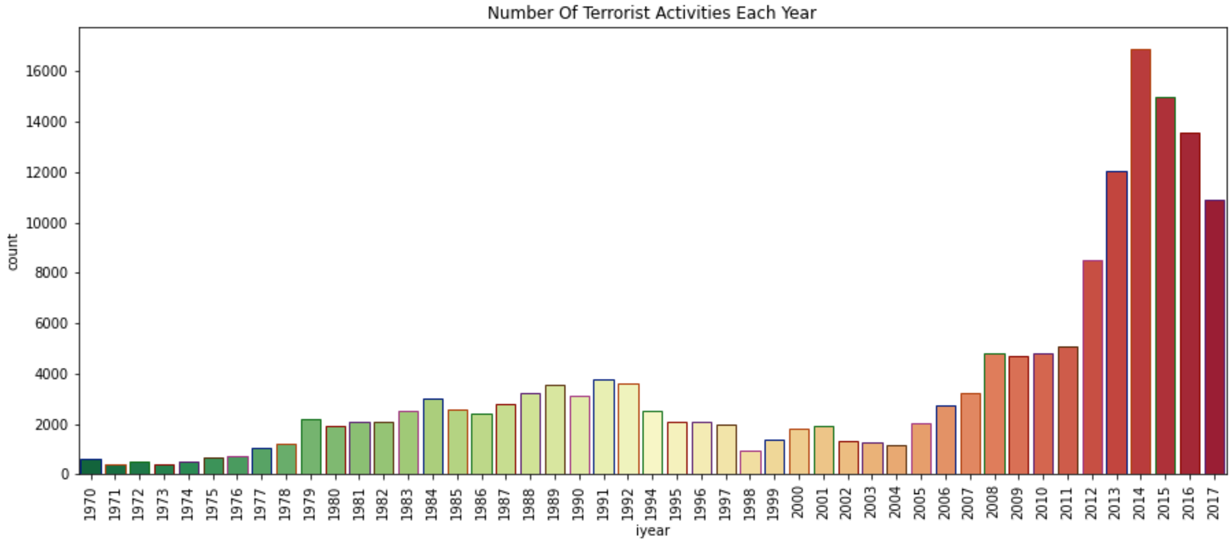
```
sklearn.metrics.accuracy_score(Y_test,Y_pred)

print(classification_report(Y_test,Y_pred))

from sklearn.metrics import confusion_matrix

cm=confusion_matrix(Y_test,Y_pred)

conf_matrix=pd.DataFrame(data=cm,columns=['Predicted:0','Predicted:1'],index=['Actual:0','Actual:1'])

plt.figure(figsize = (8,5))

sns.heatmap(conf_matrix, annot=True,fmt='d',cmap="YlGnBu")

b, t = plt.ylim()

b += 0.5

t -= 0.5

plt.ylim(b, t)

plt.show()

name='Random Forest'

kfold = KFold(shuffle=True,n_splits=5,random_state=1)

cv_results = cross_val_score(model, X_train,Y_train,cv=kfold, scoring='f1_weighted')

print("%s: %f (%f)" % (name, np.mean(cv_results),np.var(cv_results,ddof=1)))

print('cross validation scores: ',cv_results)

print('Bias error: ',np.mean((1-cv_results)*100))

print('variance error: ',np.var((1-cv_results)*100,ddof=1))

from sklearn.neighbors import KNeighborsClassifier

model= KNeighborsClassifier(n_neighbors=3)

model.fit(X_train,Y_train)
```

```
Y_pred=model.predict(X_test)

sklearn.metrics.accuracy_score(Y_test,Y_pred)

print(classification_report(Y_test,Y_pred))

from sklearn.metrics import confusion_matrix

cm=confusion_matrix(Y_test,Y_pred)

conf_matrix=pd.DataFrame(data=cm,columns=['Predicted:0','Predicted:1'],index=['Actual:0','Actual:1'])

plt.figure(figsize = (8,5))

sns.heatmap(conf_matrix, annot=True,fmt='d',cmap="YlGnBu")

b, t = plt.ylim()

b += 0.5

t -= 0.5

plt.ylim(b, t)

plt.show()

name='KNN'

kfold = KFold(shuffle=True,n_splits=5,random_state=1)

cv_results = cross_val_score(model, X_train,Y_train,cv=kfold, scoring='f1_weighted')

print("%s: %f (%f)" % (name, np.mean(cv_results),np.var(cv_results,ddof=1)))

print('cross validation scores: ',cv_results)

print('Bias error: ',np.mean((1-cv_results)*100))

print('variance error: ',np.var((1-cv_results)*100,ddof=1))

from sklearn.naive_bayes import GaussianNB

model = GaussianNB()
```

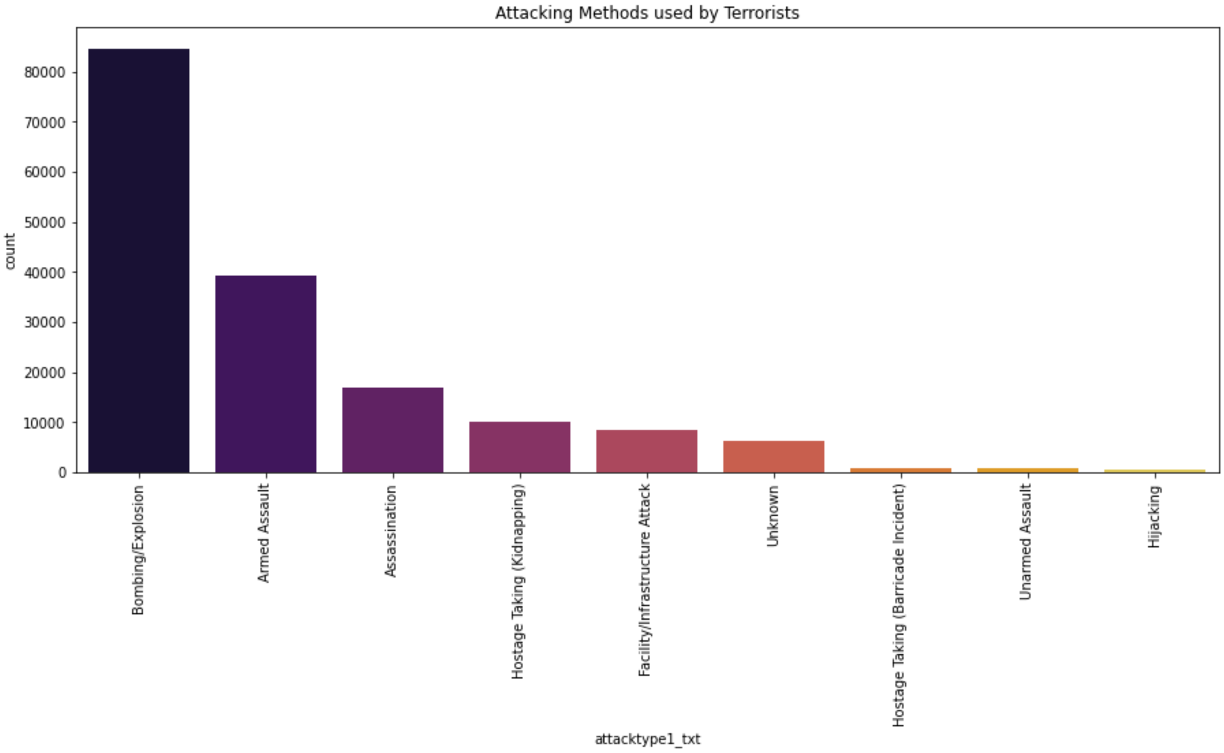
```
model.fit(X_train,Y_train)
Y_pred=model.predict(X_test)
sklearn.metrics.accuracy_score(Y_test,Y_pred)
print(classification_report(Y_test,Y_pred))
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(Y_test,Y_pred)
conf_matrix=pd.DataFrame(data=cm,columns=['Predicted:0','Predicted:1'],index=['Actual:0','Actual:1'])
plt.figure(figsize = (8,5))
sns.heatmap(conf_matrix, annot=True,fmt='d',cmap="YlGnBu")
b, t = plt.ylim()
b += 0.5
t -= 0.5
plt.ylim(b, t)
plt.show()
name='Naive Byes Classification'
kfold = KFold(shuffle=True,n_splits=5,random_state=1)
cv_results = cross_val_score(model, X_train,Y_train,cv=kfold, scoring='f1_weighted')
print("%s: %f (%f)" % (name, np.mean(cv_results),np.var(cv_results,ddof=1)))
print('cross validation scores: ',cv_results)
print('Bias error: ',np.mean((1-cv_results)*100))
print('variance error: ',np.var((1-cv_results)*100,ddof=1))
```

Chapter 6: Output Screenshots

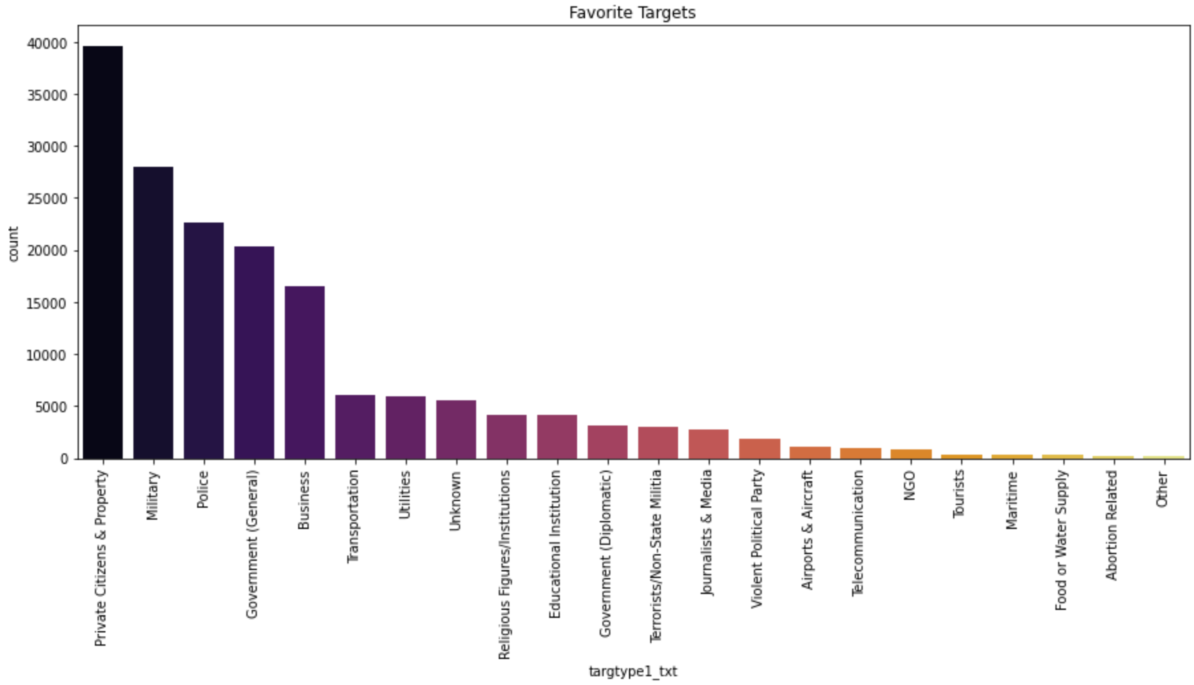
6. OUTPUT



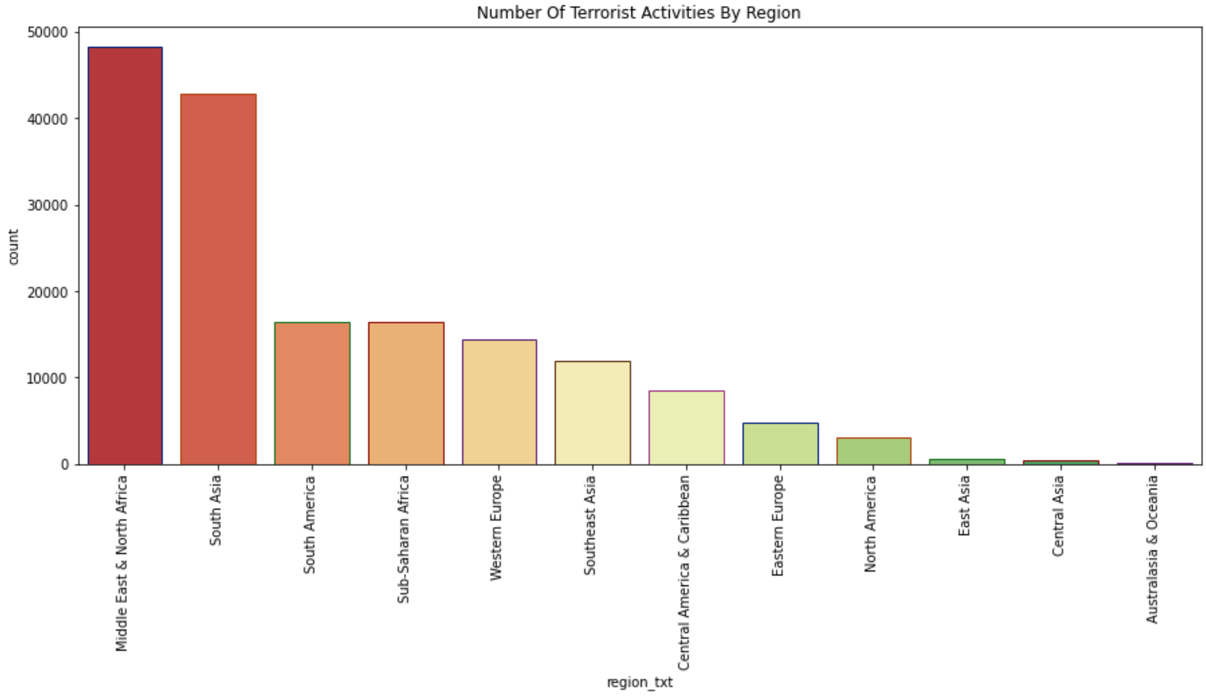
From the above graph we know that, Highest attacks were recorded in 2014



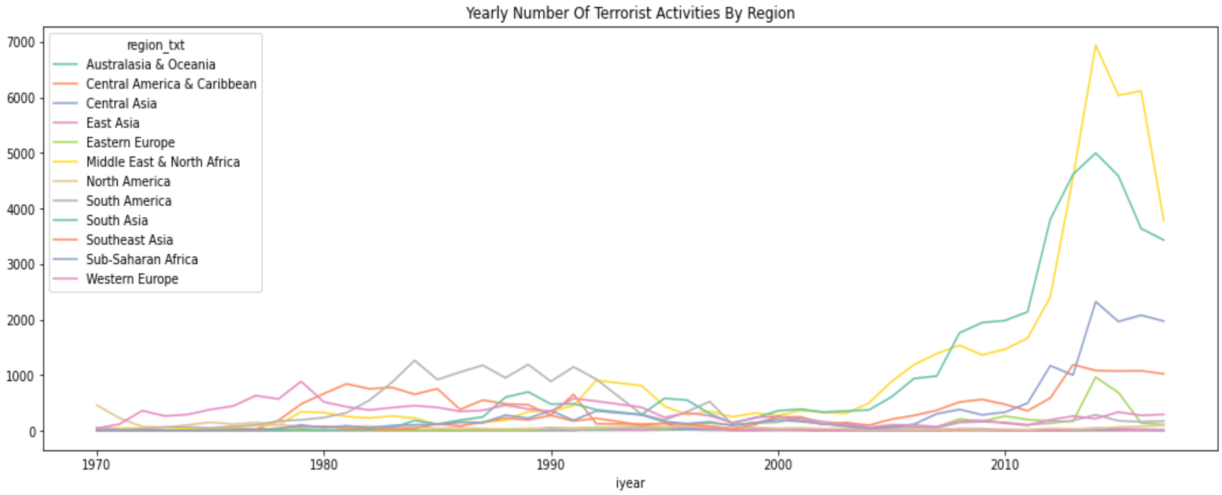
This graph represents the various attacking methods used by terrorists and the most common were bombing and explosion



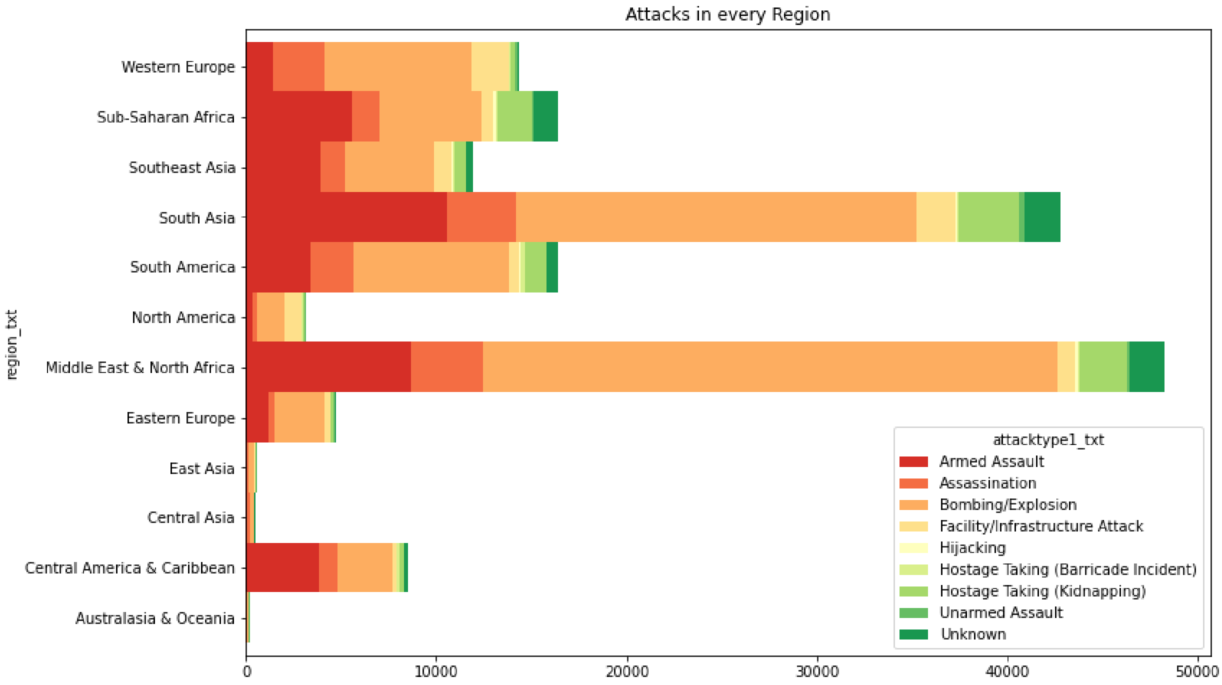
From the above graph, it can be seen that the Favorite Targets of the perpetrators are: Private Citizens and Property, Military and the Police.



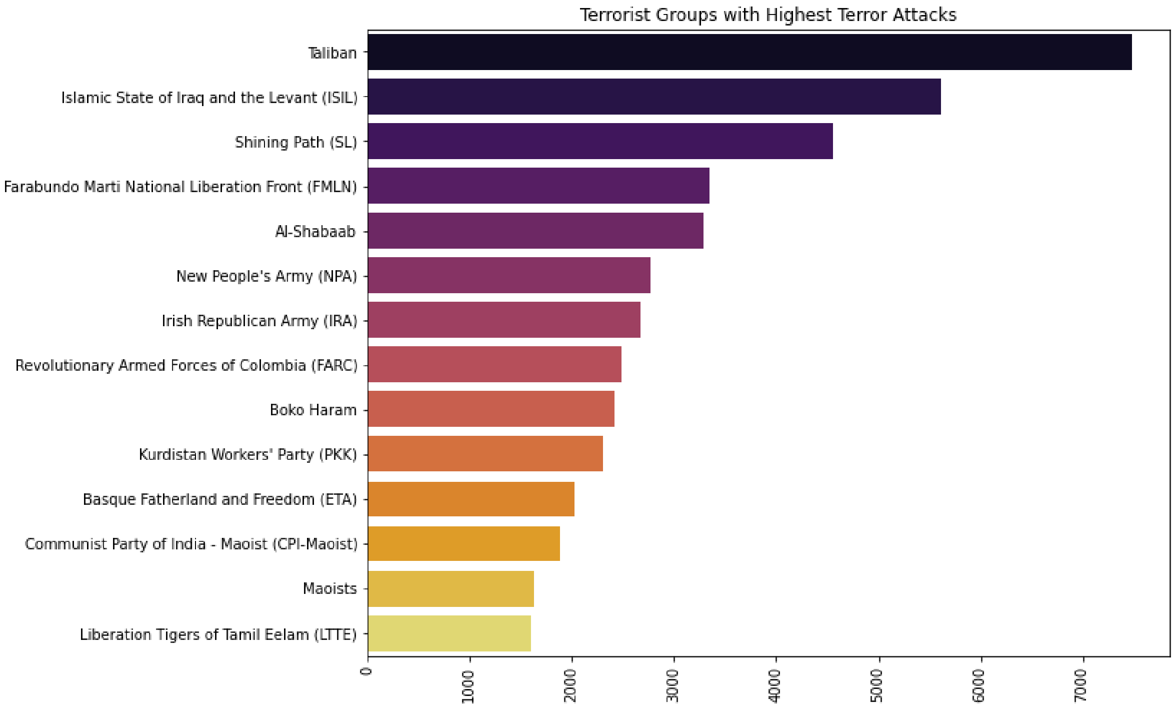
The above graph shows that the greatest number of insurgencies are in: Middle East and North Africa, South Asia, South America.



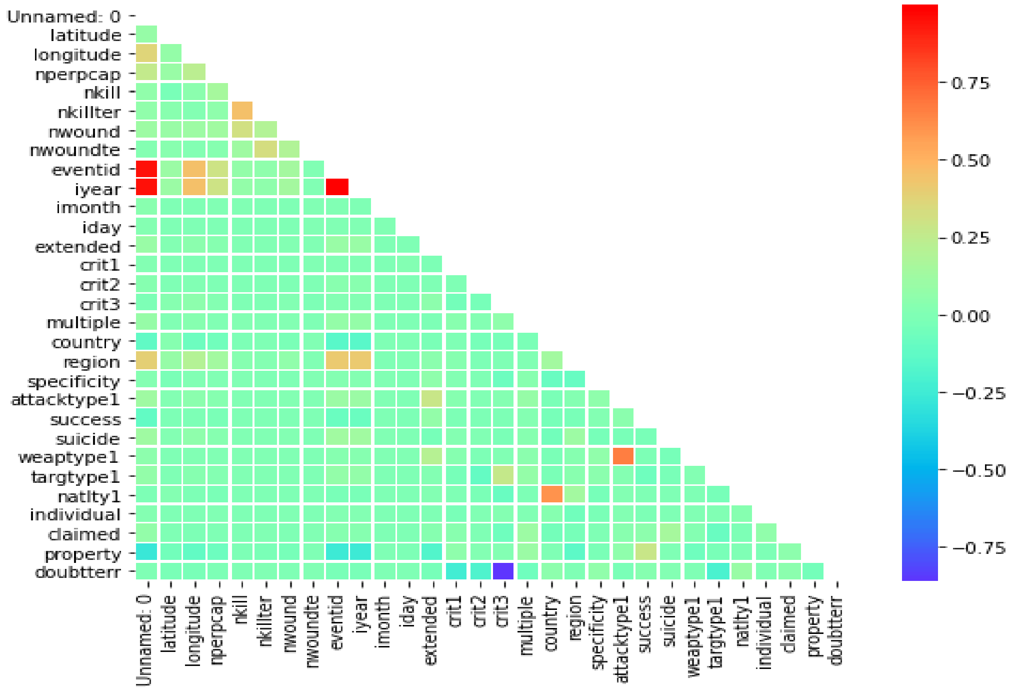
The figure above shows that the region with highest Attacks throughout years are in Middle East and North Africa.



Attacks are mostly occurring in Middle East and North Africa and the most common attack type was bombing and explosion.



The terrorist Groups that have led to the greatest number of terrorist activities are: Taliban, Islamic State of Iraq and Levant and Shining Path.



In this visual, correlation coefficients are colored according to the value. We can see that: nkillter, nwound, nwoundte are having high correlation amongst them, hence we can call them multicollinear.

Chapter 7: System Testing

7. SYSTEM TESTING

1. Logistic Regression:

After fitting training data to the logistic regression, the accuracy is: 97.41%. These are the precision recall and F1 scores

```
              precision    recall  f1-score   support

         0.0       0.97      1.00      0.98     23103
         1.0       1.00      0.84      0.91      4403

    accuracy                           0.97     27506
   macro avg       0.98      0.92      0.95     27506
weighted avg       0.97      0.97      0.97     27506
```

Cross Validation Scores: n folds (5)

Logistic Regression: 0.972428 (0.000001)

cross validation scores: [0.97189059 0.97246359 0.9733996 0.97094064 0.97344414]

Bias error: 2.7572287760646823

variance error: 0.011197154817155698

2. Decision Tree Classification:

After fitting training data to the logistic regression, the accuracy is: 95.41%. These are the precision recall and F1 scores

```
              precision    recall  f1-score   support

         0.0       0.98      0.97      0.97     23103
         1.0       0.85      0.87      0.86      4403

    accuracy                           0.95     27506
   macro avg       0.91      0.92      0.92     27506
weighted avg       0.95      0.95      0.95     27506
```

Cross Validation Scores: n folds (5)

Decision Tree: 0.951622 (0.000002)

cross validation scores: [0.9505587 0.95310809 0.95012933 0.95174194 0.95257313]

Bias error: 4.837776259695552

variance error: 0.016215431419625604

3. Random Forest Classification:

After fitting training data to the logistic regression, the accuracy is: 97.44%. These are the precision recall and F1 scores

```
              precision    recall  f1-score   support

         0.0       0.97      1.00      0.98     23103
         1.0       0.99      0.85      0.91      4403

    accuracy                           0.97     27506
   macro avg       0.98      0.92      0.95     27506
weighted avg       0.97      0.97      0.97     27506
```

Cross Validation Scores: n folds (5)

Random Forest: 0.972964 (0.000001)

cross validation scores: [0.97203847 0.97266891 0.97418606 0.97171296 0.9742161]

Bias error: 2.7035500506770527

variance error: 0.013924947229240883

4. K Nearest Neighbor:

After fitting training data to the logistic regression, the accuracy is: 97.06%. These are the precision recall and F1 scores.

```
              precision    recall  f1-score   support

         0.0       0.97      1.00      0.98     23103
         1.0       0.97      0.84      0.90      4403

    accuracy                           0.97     27506
   macro avg       0.97      0.92      0.94     27506
weighted avg       0.97      0.97      0.97     27506
```

Cross Validation Scores: n folds (5)

KNN: 0.969297 (0.000001)

cross validation scores: [0.9695715 0.96854872 0.97020054 0.96796192 0.97020048]

Bias error: 3.0703369015499327

variance error: 0.010125924035398883

5. Naïve Bayes Classification:

After fitting training data to the logistic regression, the accuracy is: 97.18%. These are the precision recall and F1 scores.

```
              precision    recall  f1-score   support

         0.0       0.97      1.00      0.98     23103
         1.0       0.98      0.85      0.91      4403

    accuracy                           0.97     27506
   macro avg       0.97      0.92      0.94     27506
weighted avg       0.97      0.97      0.97     27506
```

Cross Validation Scores: n folds (5)

Naïve Bayes: 0.969483 (0.000001)

cross validation scores: [0.96906029 0.96969223 0.97052743 0.96815993 0.96997567]

Bias error: 3.0516889775756946

variance error: 0.008266294821492123

Chapter 8: Contribution

8. CONTRIBUTION

This project entirely consisted of 7 phases which were: Understanding the problem, data acquisition, data pre-processing, exploratory data analysis, data modelling, data evaluation and documentation.

Since, the project was an individual project therefore, starting from the very first phase to the documentation of the project, whole project has been solely done by me.

For a smooth flow of this project, I had initially divided the whole project into certain stages. The first week was for identifying the business case and understanding the problem. Week-2 involved data collection and ingestion. In Week 3, data cleaning and preparation of the collected data was performed. Week 4 involved the proper Analysis of data which is better known as EDA or exploratory Data Analysis. The next few weeks i.e. in Week 5-6 various Supervised Classification Models were performed on the data in order get the most suitable model for the data. In Week 7-8 the models were evaluated and validated plus, the documentation of the project was done.

9. CONCLUSION

Through visualization of the data there were few observations that were found prominent to throw light on and those are:

1. Highest number of terrorist attacks were recorded in 2014
2. There are various attacking methods used by terrorists but, the most common are bombing and explosion
3. It was observed that the Favorite Targets of the perpetrators are: Private Citizens and Property, Military and the Police.
4. The greatest number of insurgencies or the highest affected areas are: Middle East and North Africa, South Asia, South America.
5. The terrorist Groups that have led to the greatest number of terrorist activities are: Taliban, Islamic State of Iraq and Levant and Shining Path.

Also, after various model training, testing and evaluation it was observed that Random Forest Classification and Logistic Regression models performed the best with an accuracy score of 97.44% and 97.41% respectively.

9.1 Limitations

1. Outlier Treatment might lead to information loss: There were extreme values in the data that significantly differ from the other values. However, in my case Outliers are very informative about the subject-area and data collection process.

2. Data transformation is not significant in my case as our model is a binary classification model and data need not always be nearly normal in order to perform parametric tests for the same.

9.2 Future Scope

- Latitude and Longitude act as important parameters to decide if an attack in a particular region can be called terrorist or not.
- Once we are aware of the Latitude and longitude of the regions that have a tendency of being attacked potentially, we can take the following steps:

1. Government can mark that area in the public map as a Red Alert and restrict public movement in that area.
2. The administration can also increase surveillance in that area and have more cameras both hidden and visible so that even if the visible cameras are dismantled, the footages can still be recorded from the hidden ones, this in turn will leads to an arrest of the suspicious individuals.
3. More security troops should be sent in that area in undercover form.
4. More hospitals and healthcare should be made available to individual’s in that area so that even if there is an attack despite the aforementioned steps, the risk of death can be reduced by proactive medical treatment and support.

10. BIBLIOGRAPHY

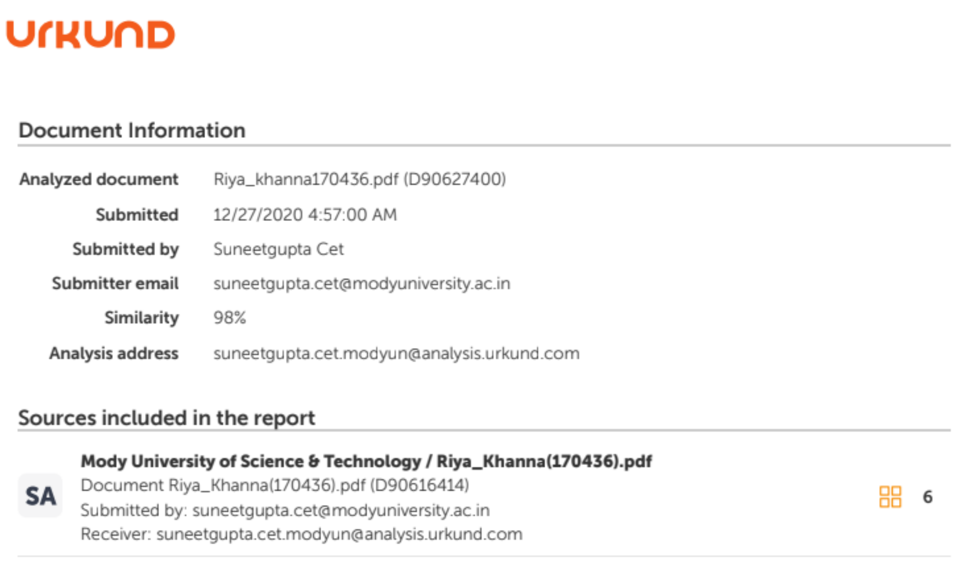
1. Kaggle. [Online] Database. https://www.kaggle.com/START-UMD/gtd.

2. [Online] March 16, 2020. https://ieeexplore.ieee.org/abstract/document/9035296.

Chapter 11: Annexures

11. Annexures

11.1 Plagiarism Report



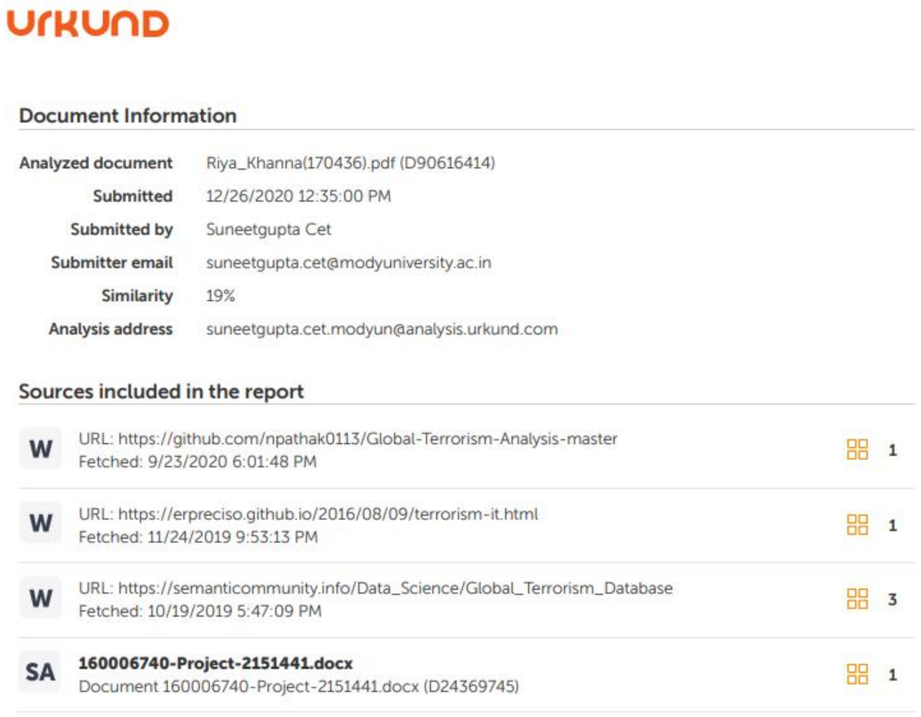
URKUND

Document Information

Analyzed document	Riya_khanna170436.pdf (D90627400)
Submitted	12/27/2020 4:57:00 AM
Submitted by	Suneetgupta Cet
Submitter email	suneetgupta.cet@modyuniversity.ac.in
Similarity	98%
Analysis address	suneetgupta.cet.modyun@analysis.urkund.com

Sources included in the report

SA **Mody University of Science & Technology / Riya_Khanna(170436).pdf**
Document Riya_Khanna(170436).pdf (D90616414)
Submitted by: suneetgupta.cet@modyuniversity.ac.in
Receiver: suneetgupta.cet.modyun@analysis.urkund.com
6



URKUND

Document Information

Analyzed document	Riya_Khanna(170436).pdf (D90616414)
Submitted	12/26/2020 12:35:00 PM
Submitted by	Suneetgupta Cet
Submitter email	suneetgupta.cet@modyuniversity.ac.in
Similarity	19%
Analysis address	suneetgupta.cet.modyun@analysis.urkund.com

Sources included in the report

W URL: https://github.com/npathak0113/Global-Terrorism-Analysis-master
Fetched: 9/23/2020 6:01:48 PM
1

W URL: https://erpreciso.github.io/2016/08/09/terrorism-it.html
Fetched: 11/24/2019 9:53:13 PM
1

W URL: https://semanticommunity.info/Data_Science/Global_Terrorism_Database
Fetched: 10/19/2019 5:47:09 PM
3

SA **160006740-Project-2151441.docx**
Document 160006740-Project-2151441.docx (D24369745)
1