INSTITUTE FOR ADVANCED COMPUTING AND SOFTWARE DEVELOPMENT AKURDI, PUNE

Documentation On

"Future Prediction on Global Terrorism"

PG-DBDA SEP 2020

Submitted By:

Group No: G-12

Akash Ganjegaonkar – 1517

Suraj Yadav – 1548

Project Guide:
Mr. Akshay Tilekar (External Guide)

Mr. Rahul Pund (Internal Guide)

Mr. Manish Bendale (Internal Guide)

Mr. Prashant Karhale
Centre Coordinator

1. Acknowledgement

First and Foremost we thank to **almighty God** for giving us the support, strength, positive spirit and talent to do this project.

"The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible".

"This work is the result of inspiration, motivation, knowledge, interest, support, guidance, cooperation and efforts by many people at different levels. We are indebted to all of them".

We would also like to take this opportunity to acknowledge the valuable contributions made by our **family members** by supporting and motivating us in every walk of life.

We are thankful to **Mr. Prashant Karhale,** Centre Co-ordinator IACSD CDAC, Akurdi Pune for providing the opportunity, infrastructure and facilities for entire work.

We would like to express our great appreciation to our project Guide **Mr. Akshay Tilekar,** Internal Project Guides **Mr. Rahul, and Mr. Manish** for their valuable and constructive suggestions during the planning and development of this project.

We also thank all staff members from the IACSD who in some way or other Helped us in completion of this project.

We cannot conclude our acknowledgement without expressing our thanks to our friends who helped us directly or indirectly during the course of this project.

Feedback for improving the contents of the report would be more than welcome

2. Introduction

2.1 Problem Statement

Future Prediction on Global Terrorism Dataset

2.2 Abstract

The Global Terrorism Database (GTD) documents more than 200,000 international and domestic terrorist attacks that occurred worldwide since 1970. With details on various dimensions of each attack, the GTD familiarizes analysts, dimensions of each attack, the GTD familiarizes analysts, policymakers, and journalists with patterns of terrorism. The GTD defines terrorist attacks as: The threatened or actual use of illegal force and violence by a non-state actor to attain a political, economic, religious , or socia goal through fear, coercion, or intimidation.

Some general findings derived from the GTD involve the nature and distribution of terrorist attacks. For example, about half of all terrorist attacks in the GTD are non-lethal, and although approximately one percent of attacks involve 25 or more fatalities, these highly lethal attacks killed more than 140,000 people in total between 1970 and 2020. The attacks in the GTD are attributed to more than 2,000 named perpetrator organizations and more than 700 additional generic groupings such as "Tamil separatists." However, two-thirds of these groups are active for less than a year and carry out fewer than four total attacks. Likewise, only 20 perpetrator groups are responsible for half of all attacks from 1970 to 2020 for which a perpetrator was identified. In general, patterns of terrorist attacks are very diverse across time and place and the GTD supports in-depth analysis of these patterns.

2.3 Product Scope

This model help us to predict top three areas where terrorism group targeted more. From this we can predict which area will be the next target according to previous data and that region provide more security to that particular areas.

2.4 Aims & Objectives

The primary goal of this project is to provide security and predict for areas which will targeted most in particular region from the past data. Before predicting the model the training of the models will be done using a bunch of different ML models and after the training is done the ML models will be compared based on their accuracy score and f1-score and the best model will be selected which will then be used to make prediction of targets of different regions.

3 Feature Description

I. GTD ID and Date

GTD ID

(eventid)
Numeric
Variable

Incidents from the GTD follow a 12-digit Event ID system.

- First 8 numbers – date recorded "yyyymmdd".
- Last 4 numbers – sequential case number for the given day (0001, 0002 etc). This is "0001" unless there is more than one case occurring on the same date.

For example, an incident in the GTD occurring on 25 July 1993 would be numbered as "199307250001". An additional GTD case recorded for the same day would be "199307250002". The next GTD case recorded for that day would be "199307250003", etc.

In rare cases, corrections to the date of a GTD attack are made. In order to maintain stable Event ID numbers, date changes are not reflected in the Event ID.

Year

(iyear)
Numeric Variable

This field contains the year in which the incident occurred. In the case of incident(s) occurring over an extended period, the field will record the year when the incident was initiated.

Month

(imonth)
Numeric
Variable

This field contains the number of the month in which the incident occurred. In the case of incident(s) occurring over an extended period, the field will record the month when the incident was initiated.

For attacks that took place between 1970 and 2011, if the exact month of the event is unknown, this is recorded as "0." For attacks that took place after 2011, if the exact month of the event is unknown, this is recorded as the midpoint of the range of possible

dates reported in source materials and the full range is recorded in the Approximate Date (*approxdate*) field below.

Day

(iday)
Numeric Variable

This field contains the numeric day of the month on which the incident occurred. In the case of incident(s) occurring over an extended period, the field will record the day when the incident was initiated.

For attacks that took place between 1970 and 2011, if the exact day of the event is unknown, this is recorded as "0." For attacks that took place after 2011, if the exact day of the event is unknown, this is recorded as the midpoint of the range of possible dates reported in source materials and the full range is recorded in the Approximate Date (approxdate) field below.

Approximate Date

(approxdate)
Text
Variable

Whenever the exact date of the incident is not known or remains unclear, this field is used to record the approximate date of the incident.

- If the day of the incident is not known, then the value for "Day" is "0". For example, if an incident occurred in June 1978 and the exact day is not known, then the value for the "Day" field is "0" and the value for the "Approximate Date" field is "June 1978".

- If the month is not known, then the value for the "Month" field is "0". For example, if an incident occurred in the first half of 1978, and the values for the day and the month are not known, then the value for the "Day" and "Month" fields will both be "0" and the value for the "Approximate Date" field is "first half of 1978."

Extended Incident?
(extended)
Categorical Variable

1 = "Yes"	The duration of an incident extended more than 24 hours.
0 = "No"	The duration of an incident extended less than 24 hours.

Date of Extended Incident Resolution

(resolution)
Numeric Date Variable

This field only applies if "Extended Incident?" is "Yes" and records the date in which the incident was resolved (hostages released by perpetrators; hostages killed; successful rescue, etc.)

II. Incident Information

Incident Summary

(summary)
Text Variable

A brief narrative summary of the incident, noting the "when, where, who, what, how, and why."

Note: This field is presently only systematically available with incidents occurring after 1997.

Inclusion Criteria
(crit1, crit2, crit3)
Categorical Variables

These variables record which of the inclusion criteria (in addition to the necessary criteria) are met. This allows users to filter out those incidents whose inclusion was based on a criterion which they believe does not constitute terrorism proper. Note that for each of the criteria variables a case is coded as "1" if source information indicates that the criterion is met and "0" if source information indicates that the criterion is not met or that there is no indication that it is met.

Criterion 1: POLITICAL, ECONOMIC, RELIGIOUS, OR SOCIAL GOAL
(CRIT1)

The violent act must be aimed at attaining a political, economic, religious, or social goal. This criterion is not satisfied in those cases where the perpetrator(s) acted out of a pure profit motive or from an idiosyncratic personal motive unconnected with broader societal change.

1 = "Yes"	The incident meets Criterion 1.
0 = "No"	The incident does not meet Criterion 1 or there is no indication that the incident meets Criterion 1.

Criterion 2: INTENTION TO COERCE, INTIMIDATE OR PUBLICIZE TO LARGER AUDIENCE(S) (CRIT2)

To satisfy this criterion there must be evidence of an intention to coerce, intimidate, or convey some other message to a larger audience (or audiences) than the immediate victims. Such evidence can include (but is not limited to) the following: pre- or post-attack statements by the perpetrator(s), past behavior by the perpetrators, or the particular nature of the target/victim, weapon, or attack type.

1 = "Yes" The incident meets Criterion 2.
0 = "No" The incident does not meet Criterion 2 or no indication.

Criterion 3: OUTSIDE INTERNATIONAL HUMANITARIAN LAW *(CRIT3)*

The action is outside the context of legitimate warfare activities, insofar as it targets non-combatants (i.e. the act must be outside the parameters permitted by international humanitarian law as reflected in the Additional Protocol to the Geneva Conventions of 12 August 1949 and elsewhere).

1 = "Yes" The incident meets Criterion 3.
0 = "No" The incident does not meet Criterion 3.

Doubt Terrorism Proper?

(doubtterr)
Categorical Variable

In certain cases there may be some uncertainty whether an incident meets all of the criteria for inclusion. In these ambiguous cases, where there is a strong possibility, but not certainty, that an incident represents an act of terrorism, the incident is included in GTD and is coded as “Yes” for this variable.

1 = "Yes" There is doubt as to whether the incident is an act of terrorism. 0 = "No" There is essentially no doubt as to whether the incident is an act of terrorism.

Note: This field is presently only systematically available with incidents occurring after 1997. If this variable was not included in the data collection process at the time the case was coded, “-9” is recorded in the database.

Alternative Designation
(alternative; alternative_txt)
Categorical Variable

This variable applies to only those cases coded as "Yes" for "Doubt Terrorism Proper?" (above). This variable identifies the most likely categorization of the incident other than terrorism.

1= Insurgency/Guerilla
Action 2= Other Crime Type
3= Inter/Intra-Group Conflict
4= Lack of Intentionality
5= State Actors (systematically coded post-2012)

Note: This field is presently only systematically available with incidents occurring after 1997.

Part of Multiple Incident

(multiple)
Categorical Variable

In those cases where several attacks are connected, but where the various actions do not constitute a single incident (either the time of occurrence of incidents or their locations are discontinuous – see Single Incident Determination section above), then "Yes" is selected to denote that the particular attack was part of a "multiple" incident.

1 = "Yes" The attack is part of a multiple incident.
0 = "No" The attack is not part of a multiple incident.

Note: This field is presently only systematically available with incidents occurring after 1997.

Related Incidents

(related)
Text
Variable

When an attack is part of a coordinated, multi-part incident the GTD IDs of the related incidents are listed here, separated by commas.

Note: This field is presently only systematically available with incidents occurring after 1997.

III. Incident Location

Country

(country; country_txt)
Categorical Variable

This field identifies the country or location where the incident occurred. Separatist regions, such as Kashmir, Chechnya, South Ossetia, Transnistria, or Republic of Cabinda, are coded as part of the "home" country.

In the case where the country in which an incident occurred cannot be identified, it is coded as "Unknown."

Note that the geo-political boundaries of many countries have changed over time. In a number of cases, countries that represented the location of terrorist attacks no longer exist; examples include West Germany, the USSR and Yugoslavia. In these cases the country name for the year the event occurred is recorded. As an example, a 1989 attack in Bonn would be recorded as taking place in West Germany (FRG). An identical attack in 1991 would be recorded as taking place in Germany.

Thus, the following change dates apply:

BREAKUP OF CZECHOSLOVAKIA:
Czech Republic – independence: 1 January 1993 Slovakia – independence: 1 January 1993

BREAKUP OF UNION OF SOVIET SOCIALIST REPUBLICS (USSR):
Russian Federation – independence: 24 August 1991
Armenia – independence: 21 September 1991
Azerbaijan – independence: 30 August 1991
Belarus – independence: 25 August 1991
Estonia – independence: 17 September 1991
Georgia – independence: 9 April 1991
Kazakhstan – independence: 16 December 1991
Kyrgyzstan – independence: 31 August 1991
Latvia – independence: 21 August 1991
Lithuania – independence: 17 September 1991
Moldova – independence: 27 August 1991
Tajikistan – independence: 9 September 1991
Turkmenistan – independence: 27 October 1991
Ukraine – independence: 24 August 1991
Uzbekistan – independence: 1 September 1991
USSR terminates: 26 December 1991 – 5 January 1992

BREAKUP OF YUGOSLAVIA:
Bosnia and Herzegovina – independence: 11 April 1992
Croatia – independence: 25 June 1991
Kosovo – UNMIK established: 10 June 1999
Macedonia – independence: 8 September 1991 Yugoslavia becomes Serbia-Montenegro: 4 February 2003
Montenegro – independence: 3 June 2006
Serbia – independence: 3 June 2006
Slovenia – independence: 25 June 1991

BREAKUP OF CZECHOSLOVAKIA:
Czech Republic – independence: 1 January 1993 Slovakia – independence: 1 January 1993

OTHER:
Eritrea – independence: 24 May 1993
Germany – unification: 3 October 1990

Country (Location) Codes
(Note: These codes are also used for the target/victim nationality fields. Entries marked with an asterisk (*) only appear as target/victim descriptors in the GTD.

4 = Afghanistan
5 = Albania
6 = Algeria
7 = Andorra
8 = Angola
10 = Antigua and Barbuda
11 = Argentina
12 = Armenia
14 = Australia
15 = Austria
16 = Azerbaijan
17 = Bahama
18 = Bahrain
19 = Bangladesh
20 = Barbados
21 = Belgium
22 = Belize
23 = Benin
24 = Bermuda*
25 = Bhutan
26 = Bolivia
28 = Bosnia-Herzegovina
29 = Botswana
30 = Brazil
31 = Brunei
32 = Bulgaria
33 = Burkina Faso
34 = Burundi
35 = Belarus
36 = Cambodia
37 = Cameroon
38 = Canada
41 = Central African Republic
42 = Chad
43 = Chile
44 = China
45 = Colombia
46 = Comoros
47 = Republic of the Congo
49 = Costa Rica
50 = Croatia
51 = Cuba
53 = Cyprus
54 = Czech Republic
55 = Denmark
56 = Djibouti
57 = Dominica
58 = Dominican Republic
59 = Ecuador
60 = Egypt
61 = El Salvador
62 = Equatorial Guinea
63 = Eritrea
64 = Estonia
65 = Ethiopia
66 = Falkland Islands
67 = Fiji
68 = Finland
69 = France
70 = French Guiana
71 = French Polynesia
72 = Gabon
73 = Gambia
74 = Georgia
75 = Germany
76 = Ghana
78 = Greece
79 = Greenland*
80 = Grenada
81 = Guadeloupe

83 = Guatemala
84 = Guinea
85 = Guinea-Bissau
86 = Guyana
87 = Haiti
88 = Honduras
89 = Hong Kong
90 = Hungary
91 = Iceland
92 = India
93 = Indonesia
94 = Iran
95 = Iraq
96 = Ireland
97 = Israel
98 = Italy
99 = Ivory Coast
100 = Jamaica
101 = Japan
102 = Jordan
103 = Kazakhstan
104 = Kenya
106 = Kuwait
107 = Kyrgyzstan
108 = Laos
109 = Latvia
110 = Lebanon
111 = Lesotho
112 = Liberia
113 = Libya
114 = Liechtenstein*
115 = Lithuania
116 = Luxembourg
117 = Macau
118 = Macedonia
119 = Madagascar
120 = Malawi
121 = Malaysia
122 = Maldives
123 = Mali
124 = Malta
125 = Man, Isle of*
126 = Marshall Islands*
127 = Martinique
128 = Mauritania
129 = Mauritius
130 = Mexico
132 = Moldova
134 = Mongolia*
136 = Morocco
137 = Mozambique
138 = Myanmar
139 = Namibia
141 = Nepal
142 = Netherlands
143 = New Caledonia
144 = New Zealand
145 = Nicaragua
146 = Niger
147 = Nigeria
149 = North Korea
151 = Norway
152 = Oman*
153 = Pakistan
155 = West Bank and Gaza Strip
156 = Panama
157 = Papua New Guinea
158 = Paraguay
159 = Peru
160 = Philippines
161 = Poland
162 = Portugal
163 = Puerto Rico*
164 = Qatar
166 = Romania
167 = Russia
168 = Rwanda
169 = Saba (Netherlands Antilles)*
173 = Saudi Arabia
174 = Senegal
175 = Serbia-Montenegro
176 = Seychelles
177 = Sierra Leone
178 = Singapore
179 = Slovak Republic
180 = Slovenia
181 = Solomon Islands
182 = Somalia
183 = South Africa
184 = South Korea
185 = Spain
186 = Sri Lanka
189 = St. Kitts and Nevis
190 = St. Lucia
192 = St. Martin*
195 = Sudan

196 = Suriname
197 = Swaziland
198 = Sweden
199 = Switzerland
200 = Syria
201 = Taiwan
202 = Tajikistan
203 = Tanzania
204 = Togo
205 = Thailand
206 = Tonga*
207 = Trinidad and Tobago
208 = Tunisia
209 = Turkey
210 = Turkmenistan
212 = Tuvalu*
213 = Uganda
214 = Ukraine
215 = United Arab Emirates
216 = Great Britain*
217 = United States
218 = Uruguay
219 = Uzbekistan
220 = Vanuatu
221 = Vatican City
222 = Venezuela
223 = Vietnam
225 = Virgin Islands (U.S.)*
226 = Wallis and Futuna
228 = Yemen
229 = Democratic Republic of the Congo
230 = Zambia
231 = Zimbabwe
233 = Northern Ireland*
235 = Yugoslavia
236 = Czechoslovakia
238 = Corsica*
334 = Asian*
347 = East Timor
349 = Western Sahara
351 = Commonwealth of Independent States*
359 = Soviet Union
362 = West Germany (FRG)
377 = North Yemen
403 = Rhodesia
406 = South Yemen
422 = International
428 = South Vietnam
499 = East Germany (GDR)
520 = Sinhalese*
532 = New Hebrides
603 = United Kingdom
604 = Zaire
605 = People's Republic of the Congo
999 = Multinational*
1001 = Serbia
1002 = Montenegro
1003 = Kosovo
1004 = South Sudan

Region
(region; region_txt)
Categorical Variable
This field identifies the region in which the incident occurred. The regions are divided into the following 12 categories, and dependent on the country coded for the case:

1 = North America
Canada, Mexico, United States

2 = Central America & Caribbean
Antigua and Barbuda, Bahamas, Barbados, Belize, Cayman Islands, Costa Rica, Cuba, Dominica, Dominican Republic, El Salvador, Grenada, Guadeloupe, Guatemala, Haiti, Honduras, Jamaica, Martinique, Nicaragua, Panama, St. Kitts and Nevis, St. Lucia, Trinidad and Tobago

3 = South America
Argentina, Bolivia, Brazil, Chile, Colombia, Ecuador, Falkland Islands, French Guiana, Guyana, Paraguay, Peru, Suriname, Uruguay, Venezuela

4 = East Asia
China, Hong Kong, Japan, Macau, North Korea, South Korea, Taiwan

5 = Southeast Asia
Brunei, Cambodia, East Timor, Indonesia, Laos, Malaysia, Myanmar, Philippines, Singapore, South Vietnam, Thailand, Vietnam

6 = South Asia
Afghanistan, Bangladesh, Bhutan, India, Maldives, Mauritius, Nepal, Pakistan, Sri Lanka

7 = Central Asia
Armenia, Azerbaijan, Georgia, Kazakhstan, Kyrgyzstan, Tajikistan, Turkmenistan, Uzbekistan

8 = Western Europe
Andorra, Austria, Belgium, Cyprus, Denmark, Finland, France, Germany, Gibraltar, Greece, Iceland, Ireland, Italy, Luxembourg, Malta, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland, United Kingdom, Vatican City, West Germany (FRG)

9 = Eastern Europe
Albania, Belarus, Bosnia-Herzegovina, Bulgaria, Croatia, Czech Republic, Czechoslovakia, East Germany (GDR), Estonia, Hungary, Kosovo, Latvia, Lithuania, Macedonia, Moldova, Montenegro, Poland, Romania, Russia, Serbia, SerbiaMontenegro, Slovak Republic, Slovenia, Soviet Union, Ukraine, Yugoslavia

10 = Middle East & North Africa
Algeria, Bahrain, Egypt, Iran, Iraq, Israel, Jordan, Kuwait, Lebanon, Libya, Morocco, North Yemen, Qatar, Saudi Arabia, South Yemen, Syria, Tunisia, Turkey, United Arab Emirates, West Bank and Gaza Strip, Western Sahara, Yemen

11 = Sub-Saharan Africa
Angola, Benin, Botswana, Burkina Faso, Burundi, Cameroon, Central African Republic, Chad, Comoros, Democratic Republic of the Congo, Djibouti, Equatorial Guinea, Eritrea, Ethiopia, Gabon, Gambia, Ghana, Guinea, Guinea-Bissau, Ivory Coast, Kenya, Lesotho, Liberia, Madagascar, Malawi, Mali, Mauritania, Mozambique, Namibia, Niger, Nigeria, People's Republic of the Congo, Republic of the Congo, Rhodesia, Rwanda, Senegal, Seychelles, Sierra Leone, Somalia, South Africa, South Sudan, Sudan, Swaziland, Tanzania, Togo, Uganda, Zaire, Zambia, Zimbabwe

12 = Australasia & Oceania
Australia, Fiji, French Polynesia, New Caledonia, New Hebrides, New Zealand, Papua New Guinea, Solomon Islands, Vanuatu, Wallis and Futuna

Province / Administrative Region /State
(provstate)
Text Variable

This variable records the name (at the time of event) of the 1st order subnational administrative region in which the event occurs.

City
(city)
Text Variable

This field contains the name of the city, village, or town in which the incident occurred. If the city, village, or town for an incident is unknown, then this field contains the smallest administrative area below provstate which can be found for the incident (e.g., district).

Vicinity
(vicinity)
Categorical Variable

1 = "Yes" The incident occurred in the immediate vicinity of the city in question.
0 = "No" The incident in the city itself

Location Details
(latitude)
Numeric Variable

This field records the latitude (based on WGS1984 standards) of the city in which the event occurred.

Longitude
(longitude)

Numeric Variable
This field records the longitude (based on WGS1984 standards) of the city in which the event occurred.

Geocoding Specificity
(specificity)
Categorical Variable

This field identifies the geospatial resolution of the latitude and longitude fields. The most specific resolution uniformly available throughout the dataset is the center of the city, village, or town in which the attack occurred.

Coordinates with greater resolution, while possible, are not systematically included in the database.

1 = event occurred in city/village/town and lat/long is for that location
2 = event occurred in city/village/town and no lat/long could be found, so coordinates are for centroid of smallest subnational administrative region identified
3 = event did not occur in city/village/town, so coordinates are for centroid of smallest subnational administrative region identified
4 = no 2nd order or smaller region could be identified, so coordinates are for center of 1st order administrative region
5 = no 1st order administrative region could be identified for the location of the attack, so latitude and longitude are unknown

IV. Attack Information

Attack Type
(attacktype1; attacktype1_txt)
Categorical Variable
This field captures the general method of attack and often reflects the broad class of tactics used. It consists of nine categories, which are defined below. Up to three attack types can be recorded for each incident. Typically, only one attack type is recorded for each incident unless the attack is comprised of a sequence of events.
When multiple attack types may apply, the most appropriate value is determined based on the hierarchy below. For example, if an assassination is carried out through the use of an explosive, the Attack Type is coded as Assassination, not Bombing/Explosion. If an attack involves a sequence of events, then the first, the second, and the third attack types are coded in the order of the hierarchy below rather than the order in which they occurred.
Attack Type Hierarchy:
Assassination
Hijacking Kidnapping
Barricade Incident
Bombing/Explosion
Armed Assault
Unarmed Assault
Facility/Infrastructure Attack
Unknown

1 = ASSASSINATION
An act whose primary objective is to kill one or more specific, prominent individuals. Usually carried out on persons of some note, such as highranking military officers, government officials, celebrities, etc. Not to include attacks on non-specific members of a targeted group. The killing of a police officer would be an armed assault unless there is reason to believe the attackers singled out a particularly prominent officer for assassination.

2 = ARMED ASSAULT
An attack whose primary objective is to cause physical harm or death directly to human beings by use of a firearm, incendiary, or sharp instrument (knife, etc.). Not to include attacks involving the use of fists, rocks, sticks, or other handheld (less-than-lethal) weapons. Also includes attacks involving certain classes of explosive devices in addition to firearms, incendiaries, or sharp instruments. The explosive device subcategories that are included in this classification are grenades, projectiles, and unknown or other explosive devices that are thrown.

3 = BOMBING/EXPLOSION
An attack where the primary effects are caused by an energetically unstable material undergoing rapid decomposition and releasing a pressure wave that causes physical damage to the surrounding environment. Can include either high or low explosives (including a dirty bomb) but does not include a nuclear explosive device that releases energy from fission and/or fusion, or an incendiary device where decomposition takes place at a much slower rate. If an attack involves certain classes of explosive devices along with firearms, incendiaries, or sharp objects, then the attack is coded as an armed assault only. The explosive device subcategories that are included in this classification are grenades, projectiles, and unknown or other explosive devices that are thrown in which the bombers are also using firearms or incendiary devices.

4 = HIJACKING An act whose primary objective is to take control of a vehicle such as an aircraft, boat, bus, etc. for the purpose of diverting it to an unprogrammed destination, force the release of prisoners, or some other political objective. Obtaining payment of a ransom should not the sole purpose of a Hijacking, but can be one element of the incident so long as additional objectives have also been stated. Hijackings are distinct from Hostage Taking because the target is a vehicle, regardless of whether there are people/passengers in the vehicle.

5 = HOSTAGE TAKING (BARRICADE INCIDENT)
An act whose primary objective is to take control of hostages for the purpose of achieving a political objective through concessions or through disruption of normal operations. Such attacks are distinguished from kidnapping since the incident occurs and usually plays out at the target location with little or no intention to hold the hostages for an extended period in a separate clandestine location.

6 = HOSTAGE TAKING (KIDNAPPING)
An act whose primary objective is to take control of hostages for the purpose of achieving a political objective through concessions or through disruption of normal operations. Kidnappings are distinguished from Barricade Incidents (above) in that they involve moving and holding the hostages in another location.

7 = FACILITY / INFRASTRUCTURE ATTACK
An act, excluding the use of an explosive, whose primary objective is to cause damage to a non-human target, such as a building, monument, train, pipeline, etc. Such attacks include arson and various forms of sabotage (e.g., sabotaging a train track is a facility/infrastructure attack, even if passengers are killed). Facility/infrastructure attacks can include acts which aim to harm an installation, yet also cause harm to people incidentally (e.g. an arson attack primarily aimed at damaging a building, but causes injuries or fatalities).

8 = UNARMED ASSAULT
An attack whose primary objective is to cause physical harm or death directly to human beings by any means other than explosive, firearm, incendiary, or sharp instrument (knife, etc.). Attacks involving chemical, biological or radiological weapons are considered unarmed assaults.

9 = UNKNOWN
The attack type cannot be determined from the available information.

Second Attack Type
(attacktype2; attacktype2_txt)
Categorical Variable
This variable utilizes the hierarchy and attack type definitions listed above.

Third Attack Type
(attacktype3; attacktype3_txt)
Categorical Variable
This variable utilizes the hierarchy and attack type definitions listed above

Successful Attack
(success)
Categorical Variable
Success of a terrorist strike is defined according to the tangible effects of the attack. Success is not judged in terms of the larger goals of the perpetrators. For example, a bomb that exploded in a building would be counted as a success even if it did not succeed in bringing the building down or inducing government repression.

The definition of a successful attack depends on the type of attack. Essentially, the key question is whether or not the attack type took place. If a case has multiple attack types, it is successful if any of the attack types are successful, with the exception of assassinations, which are only successful if the intended target is killed.

1 = "Yes" The incident was successful.
0 = "No" The incident was not successful.

ASSASSINATION

In order for an assassination to be successful, the target of the assassination must be killed. For example, even if an attack kills numerous people but not the target, it is an unsuccessful assassination.

ARMED ASSAULT

An armed assault is determined to be successful if the assault takes place and if a target is hit (including people and/or property). Unsuccessful armed assaults are those in which the perpetrators attack and do not hit the target. An armed assault is also unsuccessful if the perpetrators are apprehended on their way to commit the Sconsidered unsuccessful if they do not detonate. The success or failure of the bombing is not based on whether it hit the intended target.

HIJACKING

A hijacking is successful if the hijackers assume control of the vehicle at any point, whereas a hijacking is unsuccessful if the hijackers fail to assume control of the vehicle. The success or failure of the hijacking is not based on whether the vehicle reached the intended destination of the hijackers.

HOSTAGE TAKING (BARRICADE INCIDENT)

A barricade incident is successful if the hostage takers assume control of the individuals at any point, whereas a barricade incident is unsuccessful if the hostage takers fail to assume control of the individuals.

HOSTAGE TAKING (KIDNAPPING)

A kidnapping is successful if the kidnappers assume control of the individuals at any point, whereas a kidnapping is unsuccessful if the kidnappers fail to assume control of the individuals.

FACILITY / INFRASTRUCTURE ATTACK

A facility attack is determined to be successful if the facility is damaged. If the facility has not been damaged, then the attack is unsuccessful.

UNARMED ASSAULT

An unarmed assault is determined to be successful there is a victim that who has been injured. Unarmed assaults that are unsuccessful are those in which the perpetrators do not injure anyone. An unarmed assault is also unsuccessful if the perpetrators are apprehended when on their way to commit the assault. To make this determination, however, there must be information to indicate that an assault was imminent.

Suicide Attack
(suicide)
Categorical Variable
This variable is coded “Yes” in those cases where there is evidence that the perpetrator did not intend to escape from the attack alive.

1 = "Yes" The incident was a suicide attack.
0 = "No" There is no indication that the incident was a suicide attack.

Additional Information and Sources

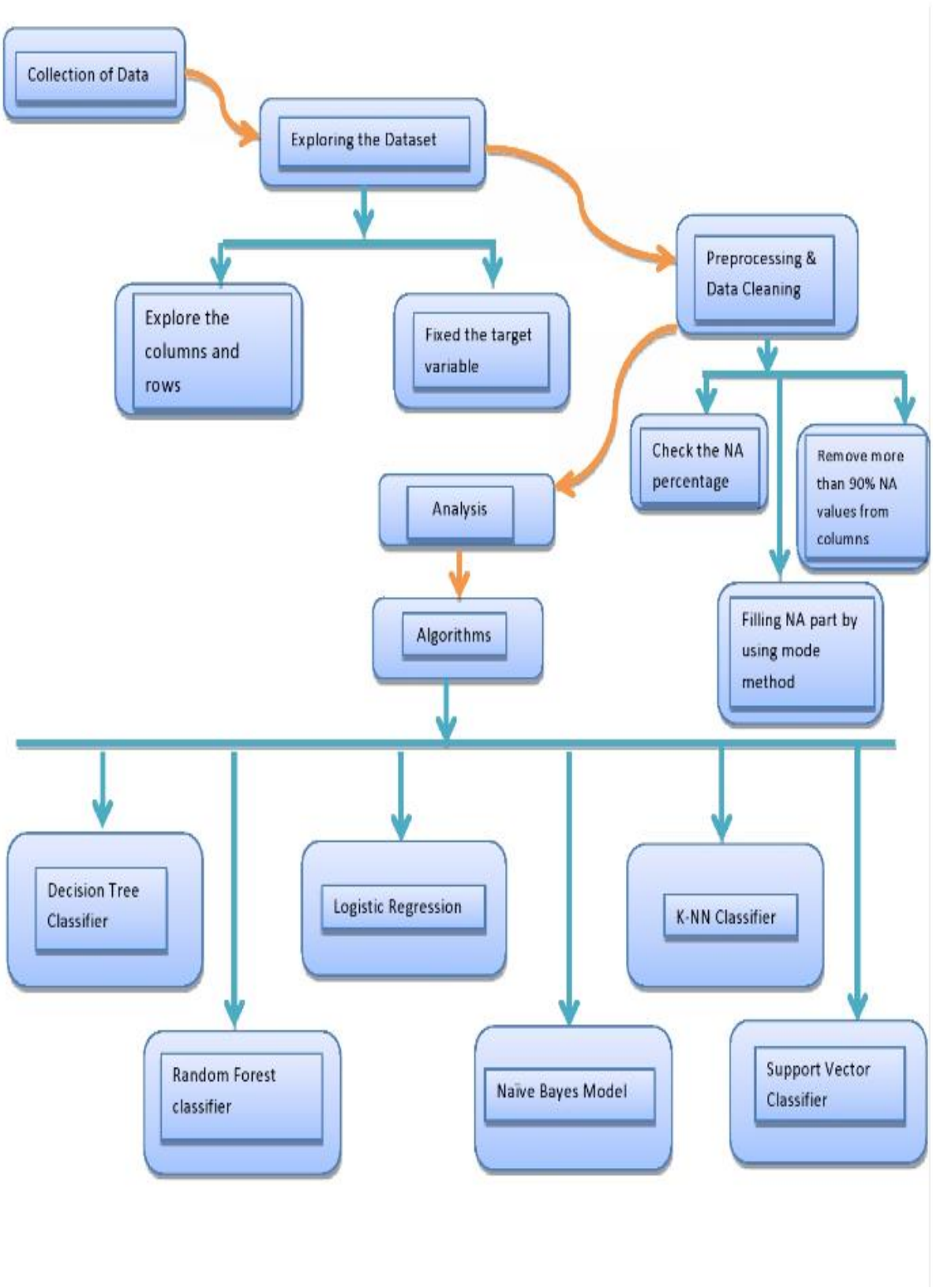
Additional Notes
(addnotes)
Text Variable

- This field is used to capture additional relevant details about the attack. It may include any of the following information:
- Additional information that could not be captured in any of the above fields, such as details about hostage conditions or additional countries hijacked vehicles were diverted to.
- Supplemental important information not specific to the particular attack, such as multiple attacks in the same area or by the same perpetrator.
- Uncertainties about the data (such as differing reports of casualty numbers, aggregated casualty numbers split across multiple incidents, or uncertainty about perpetrators responsible).
- Unusual factors, such as a shift in tactics, the reappearance of an organization, the emergence of a new organization, an attack carried out on a historical date, or an escalation of a violent campaign.
- The fate (legal, health, or otherwise) of either victims or perpetrators where this is mentioned in GTD source documents.
- In addition, the instructions for several fields listed above have specific indications for placing additional information in the "Additional Notes" field, as needed:

- Specific Target/Victim
  - If the Target/Victim is multiple victims (e.g., in a kidnapping or assassination), up to three names are recorded in the "Specific Target/Victim" field, with remaining names recorded in the "Additional Notes" field.
- Perpetrator Individual(s)' Name(s)
  - Names of individuals identified as planners, bomb-makers, etc., who are indirectly involved in an attack, may recorded in the "Additional Notes" field.
- Mode for Claim of Responsibility
  - If greater detail is needed than provided for the "Mode for Claim of Responsibility" field (for instance, a particularly novel or strange mode is used) this information may be captured in the "Additional Notes" field.
- Kidnapping/Hostage Outcome
  - If greater detail is available than the Kidnapping/Hostage Outcome field allows, then further details about the fate of hostages/kidnapped may be recorded in the "Additional Notes" field.

3.1 Work Flow Diagram

The diagram below shows the workflow of this project.



3.2 Data Preprocessing and Cleaning:

3.2.1 Data Cleaning:

The data can have many irrelevant, missing parts. To handle this part, data cleaning is done. First we check the null values in the data, after that we calculate the percentage of the missing data and remove the columns which are having more than ninety percent of the null data. We fill the remaining null values by using mode method. Then we check the standard deviation of the data after filling the null values. Then we remove unwanted string columns which are not affecting the data . As our data is already done with label encoding ,we use that label encoded column in our next performance.

We decided the target variable and finalize to go with top three categories as our target variables having 22 such categories so we decided top three which are as follows one is private citizens and property, second is Military and third is Police. By this way we clean the data and finalize the target variable.

3.3 Exploratory Data Analysis

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

Following are some plots we used to extract some useful information

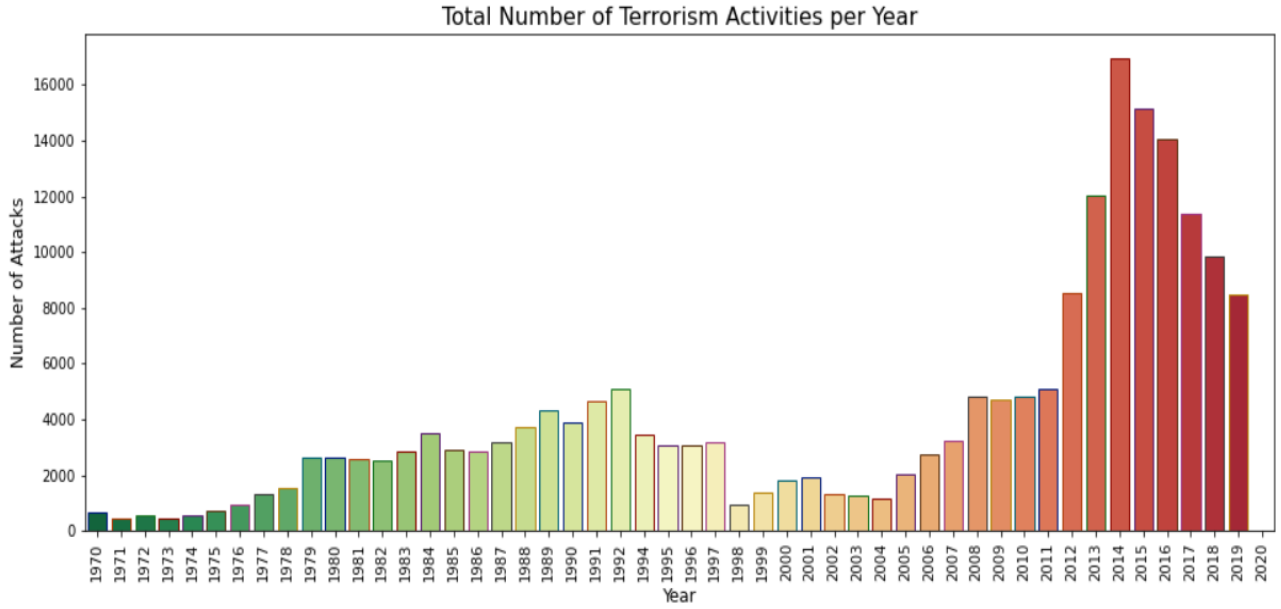


Fig:1 Total Number of Terrorism Activities per Year

fig1. In this visualization , we have used a 'count-plot' to represent the total number of terrorist activities taken place yearly-wise , between the period of 1970-2020.

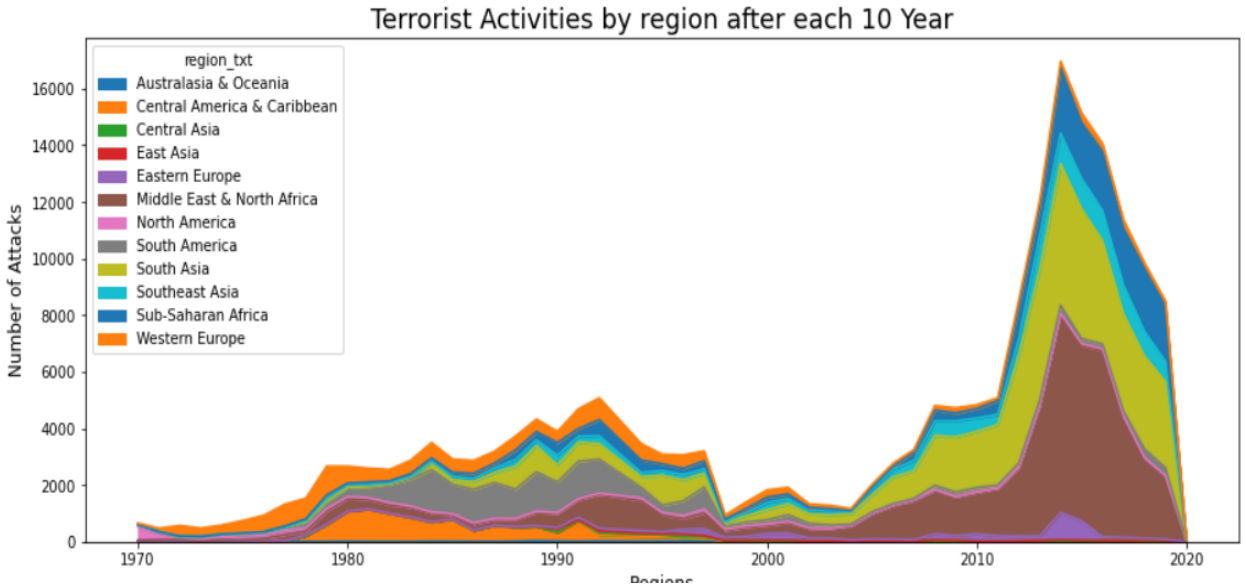


Fig:2 Terrorist Activities by region after each 10 Year

Fig2. In this visualization , we have used an 'area-chart' to represent total number of terrorist activities taken place in various regions like East-Asia , Central-Asia , North-America , etc. in the time-span of every decade(10 Years).

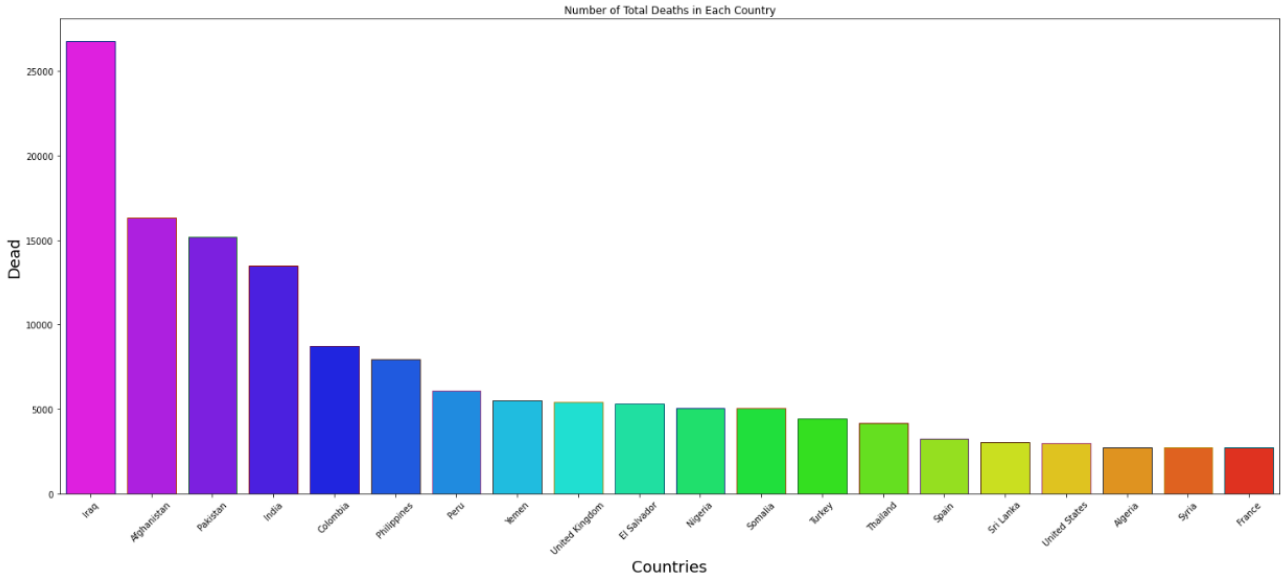


Fig:3 Number of Total Deaths in Each Country

Fig3. In this visualization , we have used a 'bar-plot' to represent the total number of deaths due to terrorist attacks in each and every country

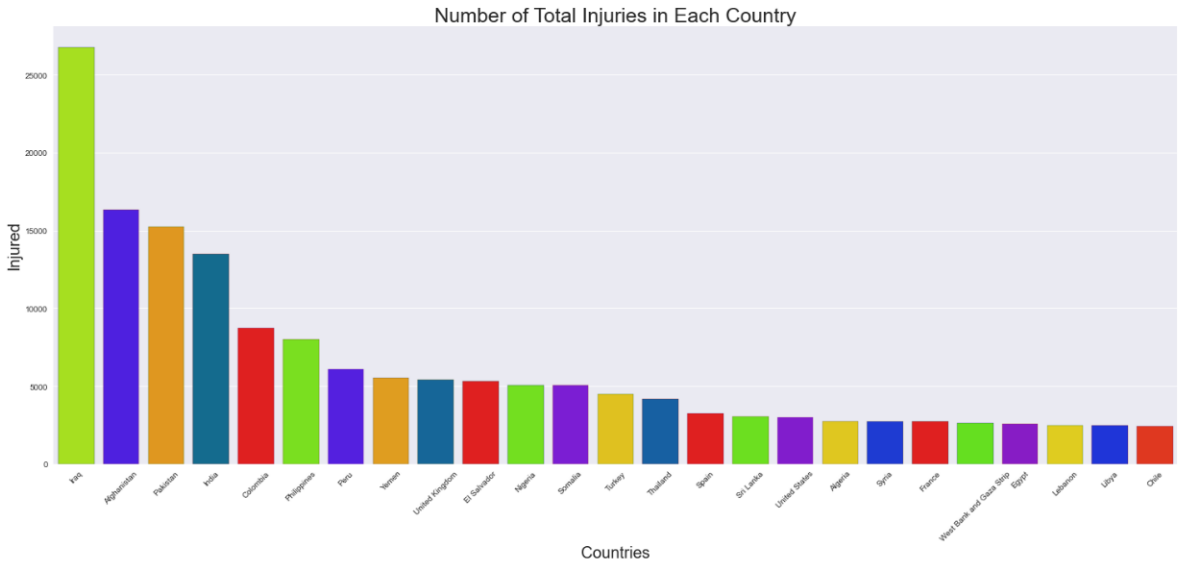


Fig:4 Number of Total Injuries in Each Country

Fig4. In this visualization , we have used a 'bar-plot' to represent the total number of injuries taken place due to terrorist attacks in each and every country

Fig: 5 Top 3 Most Targeted Areas

Fig5 . In this visualization , we have used a 'pie-chart' to represent the Top 3 most preferable areas to attack by the perpetrators and the areas where maximum security should be needed.

3.4 Model Building:

3.4.1 Train/Test split:

One important aspect of all machine learning models is to determine their accuracy. Now, in order to determine their accuracy, one can train the model using the given dataset and then predict the response values for the same dataset using that model and hence, find the accuracy of the model. A better option is to split our data into two parts: first one for training our machine learning model, and second one for testing our model.

- Split the dataset into two pieces: a training set and a testing set.
- Train the model on the training set.
- Test the model on the testing set, and evaluate how well our model did.

Advantages of train/test split:

- Model can be trained and tested on different data than the one used for training.
- Response values are known for the test dataset, hence predictions can be evaluated
- Testing accuracy is a better estimate than training accuracy of out-of-sample performance.

Machine learning consists of algorithms that can automate analytical model building. Using algorithms that iteratively learn from data, machine learning models facilitate computers to find hidden insights from Big Data without being explicitly programmed where to look.
We have used the following three algorithms to build predictive model.

3.5 Algorithms:

3.5.1 Decision Tree Classifier:-

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

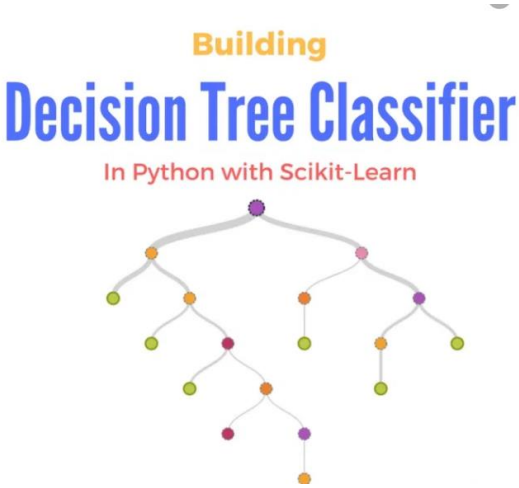


Fig:6 Decision Tree Classifier diagram

DecisionTreeClassifier

```
In [115]: from sklearn.tree import DecisionTreeClassifier
          from sklearn import tree
          model = tree.DecisionTreeClassifier(max_depth=10,splitter='best',max_features=5,min_samples_split=3,min_samples_leaf=1)

In [116]: diag=model.fit(X_train_impute,Y_train_impute)
          diag

Out[116]: DecisionTreeClassifier(max_depth=10, max_features=5, min_samples_split=3)

In [117]: model.score(X_train_impute,Y_train_impute)

Out[117]: 0.8181730665094283
```

Fig: 7 Decision Tree Classifier code

confusion matrix

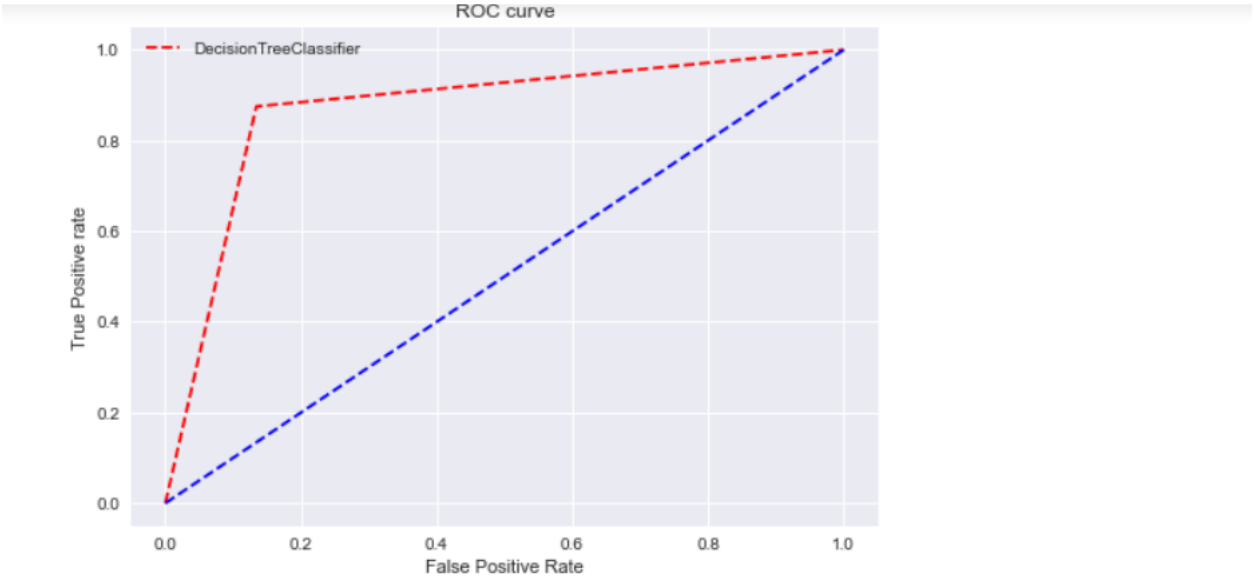
```
In [120]: from sklearn.metrics import classification_report, confusion_matrix
          print(confusion_matrix(Y_test,pred))
          print(classification_report(Y_test,pred))

[[3484  401 1439]
 [ 245 4781 1100]
 [ 484  257 8585]]
              precision    recall  f1-score   support

           3       0.83      0.65      0.73      5324
           4       0.88      0.78      0.83      6126
          14       0.77      0.92      0.84      9326

    accuracy                           0.81     20776
   macro avg       0.83      0.79      0.80     20776
weighted avg       0.82      0.81      0.81     20776
```

Fig:8 confusion matrix for DT



log-loss

```
1]: from sklearn.metrics import log_loss
    l = log_loss(Y_test,pred_prob)
    print(l)
```

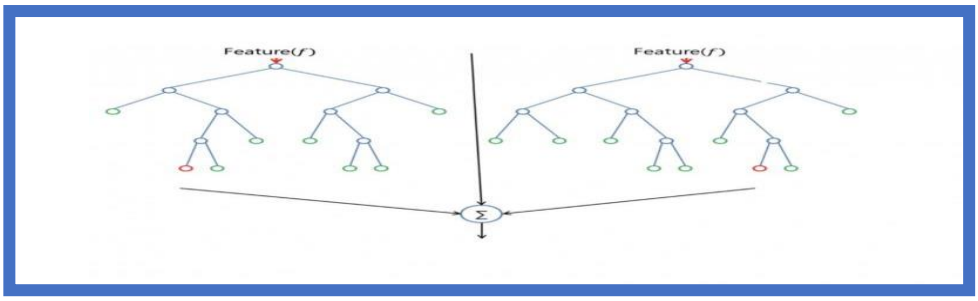
```
4.490040931338389
```

Fig: 9 log-loss and Roc curve for DT

3.5.2 Random Forest Classifier:

Random forest is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees, usually trained with the "bagging" method. The general idea of the bagging method is that a combination of learning models increases the overall result. Put simply: random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

Fig: 10 Working diagram of Random Forest



Random Forest for Classification

```
In [152]: #Random Forest for Classification
          from sklearn.ensemble import RandomForestClassifier
          model_rand = RandomForestClassifier(random_state=7)

In [153]: from sklearn.ensemble import RandomForestClassifier
          model_Class = RandomForestClassifier(n_estimators=10,criterion='gini',max_samples=35,min_samples_split=3,min_samples_leaf=1)
          model_Class.fit(X_train_impute, Y_train_impute)

Out[153]: RandomForestClassifier(max_samples=35, min_samples_split=3, n_estimators=10)

In [154]: model_Class.score(X_test,Y_test)

Out[154]: 0.8227281478629187
```

Fig: 11 Random Forest code

confusion matrix

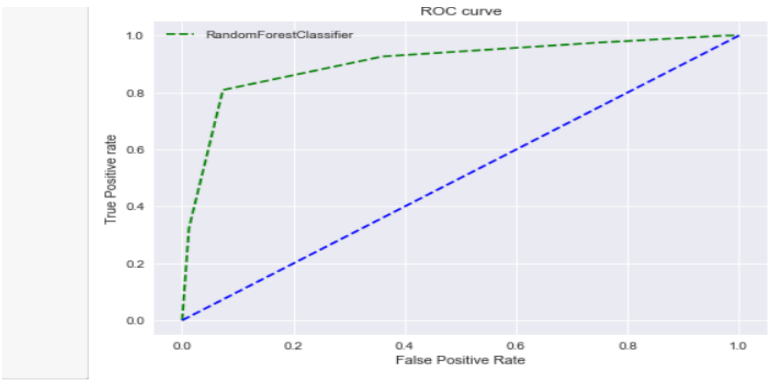
```
In [158]: from sklearn.metrics import classification_report, confusion_matrix
          print(confusion_matrix(Y_test,pred2))
          print(classification_report(Y_test,pred2))

[[2913 2226  185]
 [ 602 5442   82]
 [  45  543 8738]]
              precision    recall  f1-score   support

           3       0.82      0.55      0.66      5324
           4       0.66      0.89      0.76      6126
          14       0.97      0.94      0.95      9326

    accuracy                           0.82     20776
   macro avg       0.82      0.79      0.79     20776
weighted avg       0.84      0.82      0.82     20776
```

Fig: 12 confusion matrix for Random forest



log-loss

```
[241]: from sklearn.metrics import log_loss
       l = log_loss(Y_test,pred_prob1)
       print(l)

0.681087861564306
```

Fig: 13 log-loss and Roc curve for RF

3.5.3 Logistic regression:

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

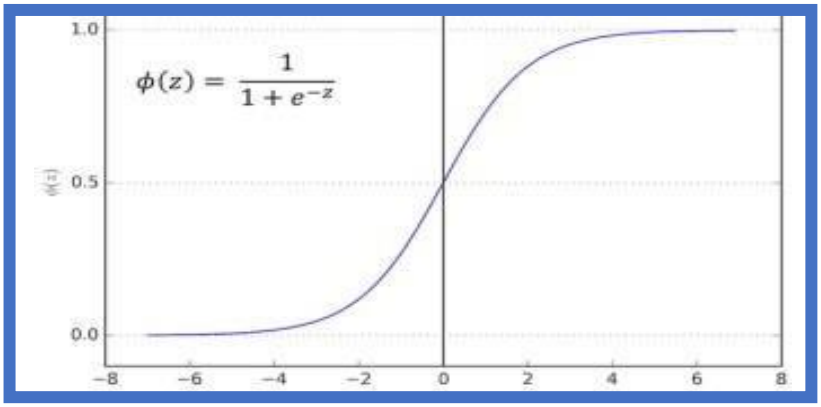


Fig: 14 Graph of Logistic Regression

Logistic Regression

In [157]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
LogisticRegr = LogisticRegression(solver='lbfgs',random_state=5,n_jobs=2,max_iter=30)
LogisticRegr.fit(X_train_impute,Y_train_impute)
Y_pred = LogisticRegr.predict(X_test)
print("Accuracy for Decision Tree classifier: ",(accuracy_score(Y_pred,Y_test)))
```

```
Accuracy for Decision Tree classifier:  0.8363977666538314
```

confusion matrix

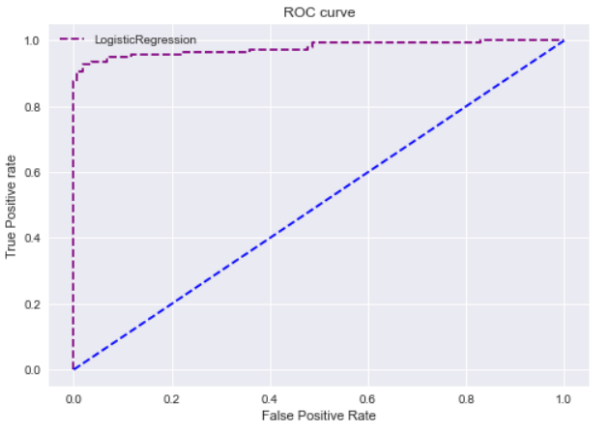
In [160]:
```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(Y_test,Y_pred))
print(classification_report(Y_test,Y_pred))
```

```
[[3292 2005   27]
 [ 981 4761  384]
 [   0    2 9324]]
              precision    recall  f1-score   support

           3       0.77      0.62      0.69      5324
           4       0.70      0.78      0.74      6126
          14       0.96      1.00      0.98      9326

    accuracy                           0.84     20776
   macro avg       0.81      0.80      0.80     20776
weighted avg       0.83      0.84      0.83     20776
```

Fig: 15 LR code and confusion matrix



log-loss

```
In [252]: from sklearn.metrics import log_loss
          l = log_loss(Y_test,pred_prob3)
          print(l)

          0.1584587778520188
```

Fig: 16 log-loss and Roc curve of LR

3.5.4 Naïve Bayes:

Naive Bayes is a kind of classifier which uses the Bayes Theorem. It predicts membership probabilities for each class such as the probability that given record or data point belongs to a particular class. The class with the highest probability is considered as the most likely class.

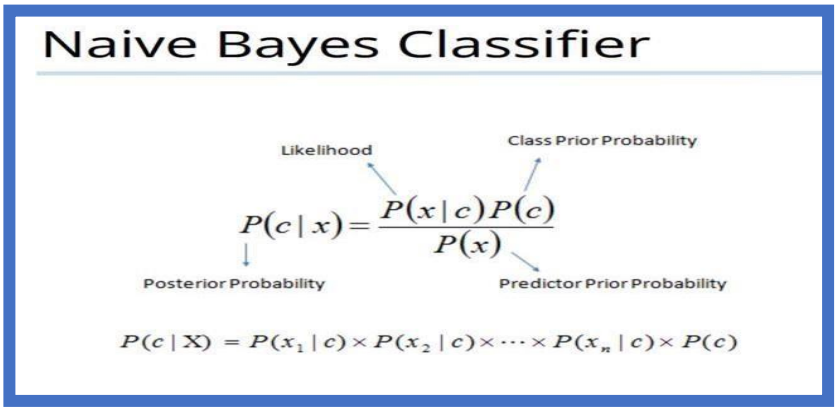


Fig: 17 NB formula

Initialize a Naive Bayes Model

In [161]:
```
from sklearn.naive_bayes import GaussianNB
# Initialize a Naive Bayes Model
model_NB = GaussianNB(priors=None, var_smoothing=1e-02) #Changed the var_smoothing
# Fit the model on given Data
model_NB.fit(X_train_impute,Y_train_impute) # patial fit
```

Out[161]:
```
GaussianNB(var_smoothing=0.01)
```

In [162]:
```
# Predict using Naive Bayes Model
Y_predict = model_NB.predict(X_test)
```

In [163]:
```
# Calculate Accuracy
acc = accuracy_score(Y_test,Y_predict)
print(acc)
```

```
0.8917982287254524
```

Fig: 18 Naïve Bayes algorithm

confusion matrix

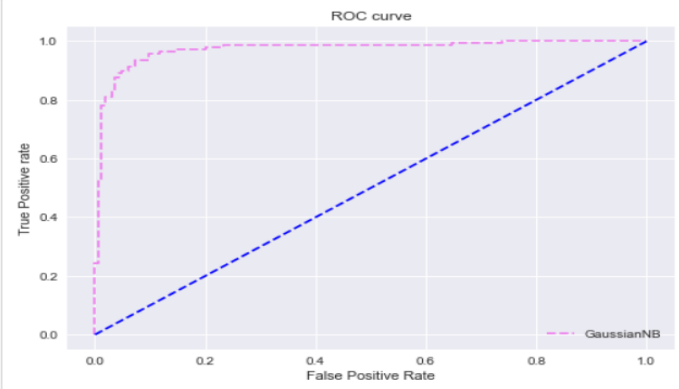
In [164]:
```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(Y_test,Y_predict))
print(classification_report(Y_test,Y_predict))
```

```
[[4379  921   24]
 [1238 4863   25]
 [   0   40 9286]]
              precision    recall  f1-score   support

           3       0.78      0.82      0.80      5324
           4       0.83      0.79      0.81      6126
          14       0.99      1.00      1.00      9326

    accuracy                           0.89     20776
   macro avg       0.87      0.87      0.87     20776
weighted avg       0.89      0.89      0.89     20776
```

Fig: 19 confusion matrix for NB



log-loss

n [268]:
```
from sklearn.metrics import log_loss
l = log_loss(Y_test,pred_prob4)
print(l)
```

```
0.2490953299921082
```

Fig: 20 Roc curve and log-loss for NB

3.5.5 K nearest neighbor:

K Nearest Neighbor Algorithm. K nearest neighbor algorithm is very simple. It works based on minimum distance from the query instance to the training samples to determine the K-nearest neighbors. The data for KNN algorithm consist of several multivariate attributes name that will be used to classify. k' in KNN is a parameter that refers to the number of nearest neighbors to include in the majority of the voting process.

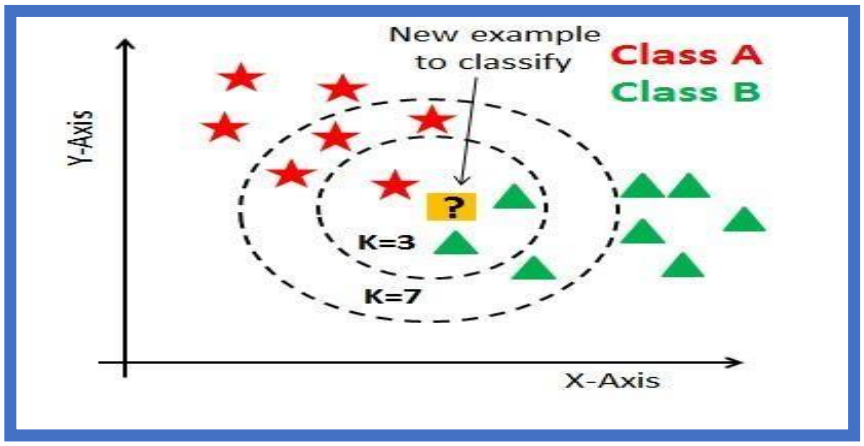


Fig: 21 Example of KNN

Fitting K-NN classifier to the training set

```
In [165]: #Fitting K-NN classifier to the training set
          from sklearn.neighbors import KNeighborsClassifier
          classifier= KNeighborsClassifier(algorithm='auto',leaf_size=30,n_neighbors=7, metric='minkowski', p=2)
          classifier.fit(X_train_impute, Y_train_impute)

Out[165]: KNeighborsClassifier(n_neighbors=7)

In [166]: #Predicting the test set result
          y_pred1= classifier.predict(X_test)

In [167]: # Calculate Accuracy
          acc = accuracy_score(Y_test,y_pred1)
          print(acc)

          0.9724201001155179
```

Fig: 22 KNN algorithm

confusion matrix

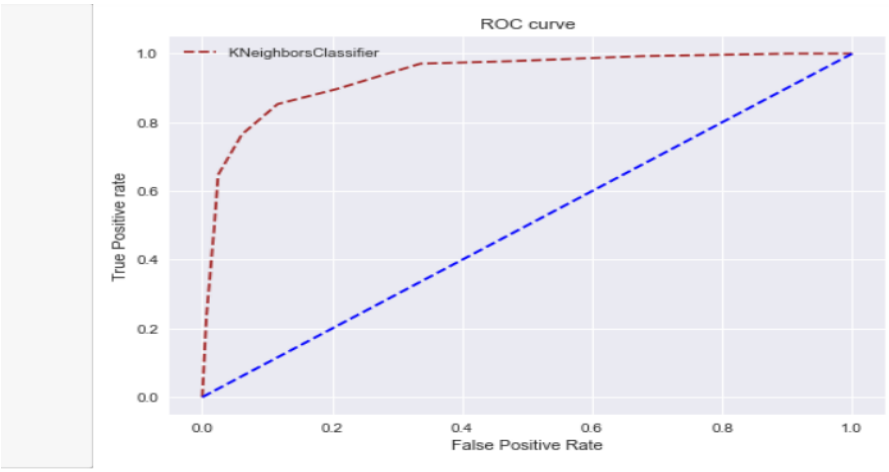
```
In [169]: from sklearn.metrics import classification_report, confusion_matrix
          print(confusion_matrix(Y_test,y_pred1))
          print(classification_report(Y_test,y_pred1))
```

```
[[5080  220   24]
 [ 303 5798   25]
 [   1    0 9325]]
              precision    recall  f1-score   support

           3       0.94      0.95      0.95      5324
           4       0.96      0.95      0.95      6126
          14       0.99      1.00      1.00      9326

    accuracy                           0.97     20776
   macro avg       0.97      0.97      0.97     20776
weighted avg       0.97      0.97      0.97     20776
```

Fig: 23 confusion matrix for KNN



log-loss

```
[274]: from sklearn.metrics import log_loss
       l = log_loss(Y_test,pred_prob5)
       print(l)
```

```
0.4572563585024048
```

Fig: 24 Roc curve and log loss for KNN

3.5.6 SVC:

The objective of a Linear SVC (Support Vector Classifier) is to fit to the data you provide, returning a "best fit" hyperplane that divides, or categorizes, your data. From there, after getting the hyperplane, you can then feed some features to your classifier to see what the "predicted" class is. This makes this specific algorithm rather suitable for our uses, though you can use this for many situations

SVM offers very high accuracy compared to other classifiers such as logistic regression, and decision trees. It is known for its kernel trick to handle nonlinear input spaces. It is used in a variety of applications such as face detection, intrusion detection, classification of emails, news articles and web pages, classification of genes, and handwriting recognition.

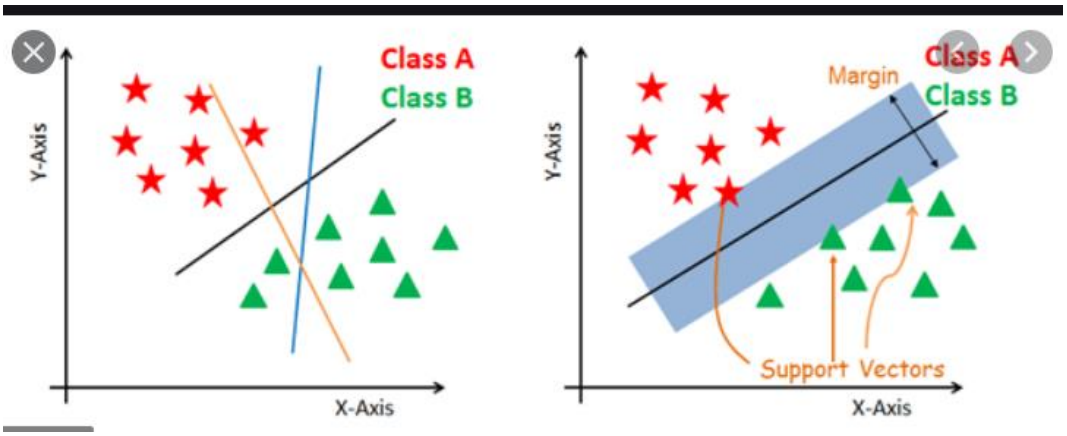


Fig:25 Diagram for svc

Support vector classifier

```
In [170]: from sklearn.svm import SVC
          model = SVC(C=0.1) # Support vector classifier

In [171]: model.fit(X_train_impute,Y_train_impute)
Out[171]: SVC(C=0.1)

In [172]: Y_predict2 = model.predict(X_test)

In [173]: # Calculate Accuracy
          acc = accuracy_score(Y_test,Y_predict2)
          print(acc)

          0.7707450904890258
```

Fig: 26 Svc algorithm

confusion matrix

```
In [174]: from sklearn.metrics import classification_report, confusion_matrix
          print(confusion_matrix(Y_test,Y_predict2))
          print(classification_report(Y_test,Y_predict2))
```

```
[[3210 2090   24]
 [2599 3502   25]
 [   1   24 9301]]
              precision    recall  f1-score   support

           3       0.55      0.60      0.58      5324
           4       0.62      0.57      0.60      6126
          14       0.99      1.00      1.00      9326

    accuracy                           0.77     20776
   macro avg       0.72      0.72      0.72     20776
weighted avg       0.77      0.77      0.77     20776
```

log-loss

```
In [281]: from sklearn.metrics import log_loss
          l = log_loss(Y_test,pred_prob)
          print(l)
```

```
2.9934032660885372
```

Fig: 27 confusion matrix and log-loss of svc

3.5.7 K-Fold:

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.

The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation.

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split.

```
In [180]: from sklearn.model_selection import KFold
          kf = KFold(n_splits=3,shuffle=False)
          kf

Out[180]: KFold(n_splits=3, random_state=None, shuffle=False)

In [181]: for train_index, test_index in kf.split([1,2,3,4,5,6,7,8,9]):
              print(train_index, test_index)

          [3 4 5 6 7 8] [0 1 2]
          [0 1 2 6 7 8] [3 4 5]
          [0 1 2 3 4 5] [6 7 8]

In [182]: def get_score(df5, X_train_impute, X_test, Y_train_impute, Y_test):
              model.fit(X_train_impute, Y_train_impute)
              return model.score(X_test, Y_test)

In [184]: from sklearn.model_selection import StratifiedKFold
          folds = StratifiedKFold(n_splits=3)

          scores_logistic = []
          scores_svm = []
          scores_rf = []

          for train_index, test_index in folds.split(X,Y):
              X_train_impute, X_test, Y_train_impute, Y_test = X[train_index],X[test_index], \
                                                               Y[train_index], Y[test_index]
              scores_logistic.append(get_score(LogisticRegression(solver='liblinear',multi_class='ovr'), X_train_impute, X_test, Y_train_in
              scores_svm.append(get_score(SVC(gamma='auto'), X_train_impute, X_test, Y_train_impute, Y_test))
              scores_rf.append(get_score(RandomForestClassifier(n_estimators=40), X_train_impute, X_test, Y_train_impute, Y_test))
```

Fig : 28 K-Fold algorithm

```
In [185]: scores_logistic
          np.average(scores_logistic)

Out[185]: 0.9210018401635168

In [187]: scores_svm
          np.average(scores_svm)

Out[187]: 0.9210018401635168

In [185]: scores_rf
          np.average(scores_rf)

Out[185]: 0.866620214507396
```

Fig: 29 Accuracy score using K-Fold

Parameter tunning using k fold cross validation

```
In [192]: scores1 = cross_val_score(RandomForestClassifier(n_estimators=5), X_train_impute,Y_train_impute, cv=10)
          np.average(scores1)

Out[192]: 0.9490728177295342

In [193]: scores2 = cross_val_score(RandomForestClassifier(n_estimators=20), X_train_impute,Y_train_impute, cv=10)
          np.average(scores2)

Out[193]: 0.9655359565807327

In [194]: scores3 = cross_val_score(RandomForestClassifier(n_estimators=30), X_train_impute,Y_train_impute, cv=10)
          np.average(scores3)

Out[194]: 0.968475802804161

In [195]: scores4 = cross_val_score(RandomForestClassifier(n_estimators=40), X_train_impute,Y_train_impute, cv=10)
          np.average(scores4)

Out[195]: 0.971506105834464
```

Fig : 30 parameter tunning using K-Fold

Validation:

This process of deciding whether the numerical results quantifying hypothesized relationships between variables, are acceptable as descriptions of the data, is known as **validation**. Generally, an error estimation for the model is made after training, better known as evaluation of residuals. In this process, a numerical estimate of the difference in predicted and original responses is done, also called the training error. However, this only gives us an idea about how well our model does on data used to train it. Now its possible that the model is underfitting or overfitting the data. So, the problem with this evaluation technique is that it does not give an indication of how well the learner will generalize to an independent/ unseen data set. Getting this idea about our model is known as Cross Validation.

Cross-validation is a technique for evaluating ML models by training several ML models on subsets of the available input data and evaluating them on the complementary subset of the data. Use cross-validation to detect overfitting, ie, failing to generalize a pattern.

In ML, you can use the k-fold cross-validation method to perform cross-validation. In k-fold cross-validation, you split the input data into k subsets of data (also known as folds). You train an ML model on all but one (k-1) of the subsets, and then evaluate the model on the subset that was not used for training. This process is repeated k times, with a different subset reserved for evaluation (and excluded from training) each time.

The following diagram shows an example of the training subsets and complementary evaluation subsets generated for each of the four models that are created and trained during a 4-fold cross-validation. Model one uses the first 25 percent of data for evaluation, and the remaining 75 percent for training. Model two uses the second subset of 25 percent (25 percent to 50 percent) for evaluation, and the remaining three subsets of the data for training, and so on.

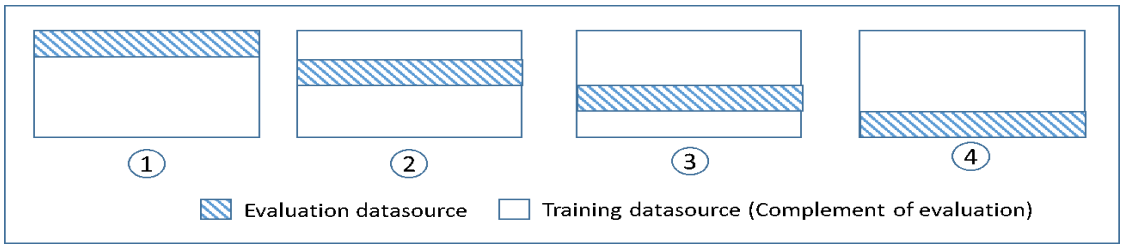


Fig: 31 cross-validation diagram

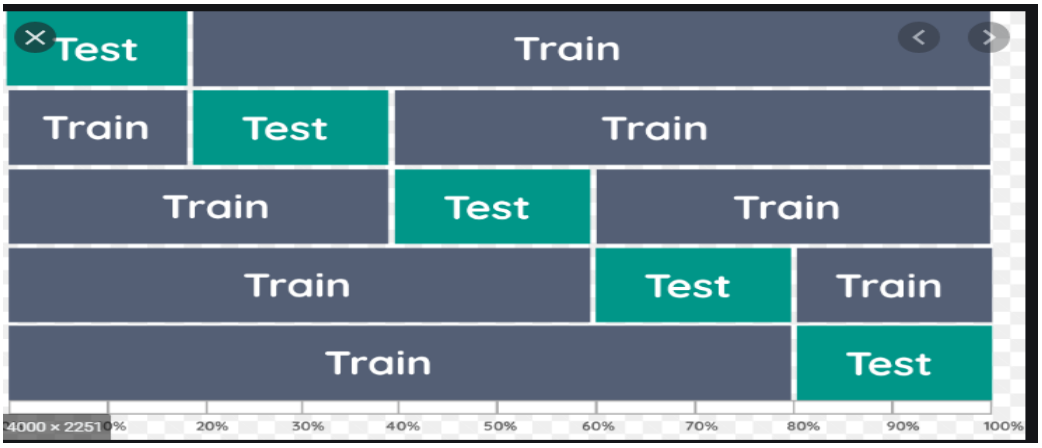


Fig: 32 K-Fold diagram

cross_val_score function

```
In [188]: from sklearn.model_selection import cross_val_score

In [189]: # Logistic regression model performance using cross_val_score
          a = cross_val_score(LogisticRegression(solver='lbfgs',multi_class='ovr',random_state=5,n_jobs=2,max_iter=30)
                                                 , X_train_impute, Y_train_impute,cv=3)
          np.average(a)
Out[189]: 0.9490297472360253

In [176]: # svm model performance using cross_val_score

In [312]: b = cross_val_score(SVC(gamma='auto'),  X_train_impute,Y_train_impute,cv=3)
          np.average(b)
Out[312]: 0.958579411369111

In [313]: # random forest performance using cross_val_score
          c = cross_val_score(RandomForestClassifier(n_estimators=40), X_train_impute,Y_train_impute,cv=3)
          np.average(c)
Out[313]: 0.9728733355342798

In [316]: # KNN perfcross_val_scorence using corss_val_score
          scores5 = cross_val_score(KNeighborsClassifier(algorithm='auto',leaf_size=30,n_neighbors=7, metric='minkowski', p=2), X_train_imp
          np.average(scores5)
Out[316]: 0.8928628199014464
```

Fig: 33 cross-validation code

3.5.8 Testing on Real-Time data

Real-time data (RTD) is information that is delivered immediately after collection. There is no delay in the timeliness of the information provided. Real-time data is often used for navigation or tracking. We created the new data file in which we randomly enter the data for the testing out algorithm.

After testing the data we get the right prediction it tells us that our algorithm predicting the right things. To check the algorithms performance this step is very important.

A3

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Year	Month	Day	Extended	Country	Region	Specificity	Vicinity	crit1	crit2	crit3	doubtterr	alternative	multiple	success	suicide	attacktype	targsubtyp	natlty1	guncertain	individual
2	2021	3	28	0	92	6	1	0	1	1	1	0	1	0	1	0	3	22	215	0	0
3																					
4																					
5																					
6																					

Fig: 34 Real-Time dataset

Testing on real-time data

```
In [175]: n = pd.read_csv("D:/DBDA/project/GTD/Real-TimeTestingData.csv" , encoding = 'latin-1' , index_col=False)

In [176]: pred_n= model_Class.predict(n)

In [177]: pred_n

Out[177]: array([3], dtype=int64)
```

Fig: 35 Testing data code

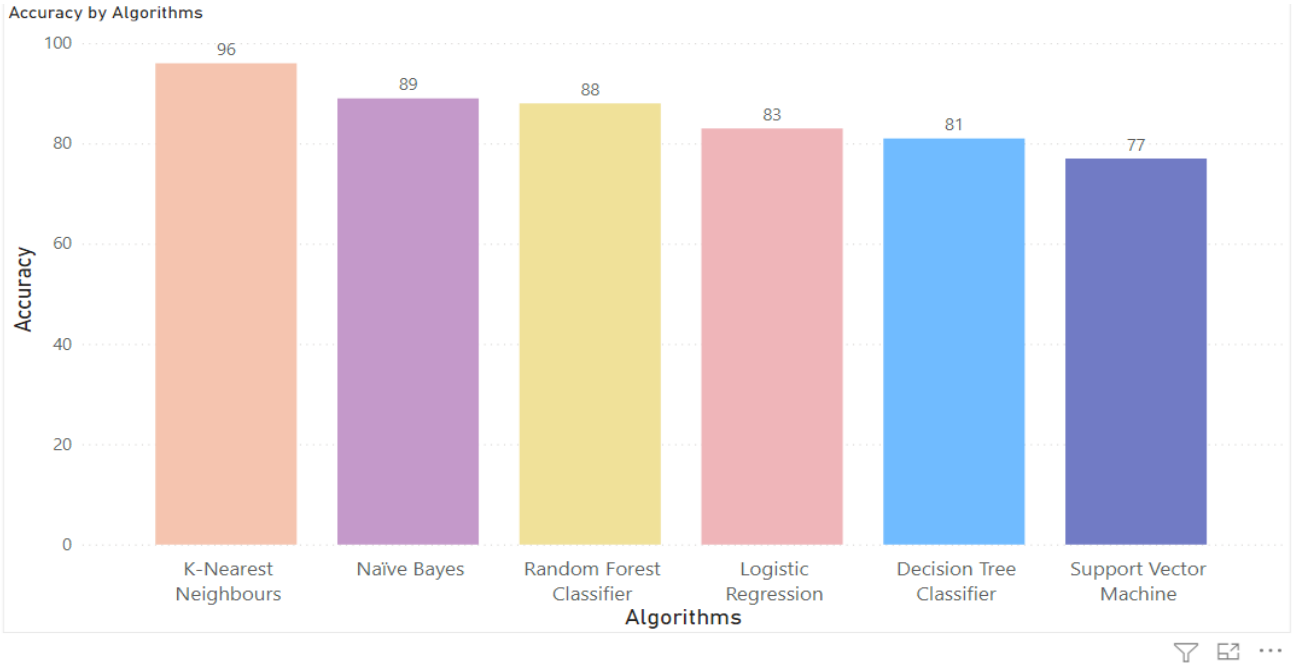


Fig: 36 Accuracy dia of all the algorithms

4. Requirements Specification

4.1 Hardware Requirement:

- 500 GB hard drive (Minimum requirement)
- 8 GB RAM (Minimum requirement)
- PC x64-bit CPU

4.2 Software Requirement:

- Windows/Mac/Linux
- Python-3.9.1
- VS Code/Anaconda/Spyder
- Python Extension for VS Code
- Libraries:
  - Numpy 1.18.2
  - Pandas 1.2.1
  - Matplotlib 3.3.3
  - Scikit-learn 0.24.1

5. Conclusion:

- This system has utilized various ML algorithms to learn and train on the basis of past data.
- And out of these several algorithms is providing the best overall accuracy as compared to any other algorithms.
- It is concluded from accuracy that is highly suitable to predict the possibilities of terrorist attacks in the future.
- We found that there are some areas which were targeted most by terrorism groups, from this algorithm , we tried to highlights those areas which need security.
- Terrorism has become a huge threat over the world. Various Machine learning system, artificial intelligence and Data-Analytics have provided us with a system to help the investigator and anti-terrorist or counter-terrorist squad to rapidly decide the most probable perpetrator responsible of a particular terrorist attack.

6. Future Scope

- In future we further mean to attempt different algorithms and methods like deep learning models and package classifiers to further improvise the accuracy of result and hence successfully predict the perpetrator with more precision and accuracy.
- Besides this in future we also intend to use web-scrapping methods and sentimental analysis to study various posts and comments on social media sites for hatred speech and text and further filter them and create a classifier to merge the current project with the social media texts.
- In future we will also going to predict all the areas which where targeted and from that particular region get the security.

7. References

- Data set from- https://start.umd.edu/research-projects/global-terrorism-database-gtd
- Kelly, Fergus (10 January 2020). *"At least 25 Niger soldiers, 63 'terrorists' killed in attack on army base in Tillaberi region"*. The Defense Post. *Retrieved 11 January 2020*.
- International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-9 Issue-1, October 2019