globsyn

Taking People To The Next Level

Globsyn finishing school, Globsyn Crystals, 1st Floor, XI – 11 and 12, Block EP, Sector V, Salt Lake, Kolkata – 700091

GLOBAL TERRORISM DATA

Group Members:

SOUMADIP MAJUMDAR, ASANSOL ENGINEERING COLLEGE, 161080120008
TUSHAR BANERJEE, ASANSOL ENGINEERING COLLEGE, 161080120009
ARPITA KARMAKAR, ASANSOL ENGINEERING COLLEGE, 161080120002
ANGSHUMAN DEY, ASANSOL ENGINEERING COLLEGE, 161080120001
BINAY KUMAR BURNWAL, ASANSOL ENGINEERING COLLEGE, 161080120003
VISHAL SHARMA,ASANSOL ENGINEERING COLLEGE,151080110124

Table of Contents

Acknowledgement

I take this opportunity to express my profound gratitude and deep regards to my faculty (TITAS ROY CHOWDHURY) for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him/her time to time shall carry me a long way in the journey of life on which I am about to embark.

I am obliged to my project team members for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

(SOUMADIP MAJUMDAR)

Project Objective

The primary project goals consist of:

As the project is based on global terrorism data analysis and prediction so with the help of it we can do the following things:

1. It is en-lighting us with the names of various terrorist groups who have attacked in the past and also have a chance to attack in the coming future.
2. This project also aims to deliver us a report of the various weapon types being used in the attacks and their devastating capacity.
3. Finally this project gives us a view and an assumption of the new weapons that can be used by various terrorist to cause damage to this world..

Project Scope

1. The system will predict us with the names of terrorist groups that can attack in the future.
2. The weapons can be understood on a before hand basis and actions can be takes to disarm them if possible.
3. The world will be enlightened with the new weapons that could be used in the future and provide the defense forces of various countries to counter attack them as they will know what can be these weapons..

Requirement Specification

- **Problem Definition**

  - Perform analysis of the data and use Native Bayes and Nearest Neighbor models to predict the following:
    1. Predict the terrorist group, given other data fields. What are the attributes that best correlate to terrorist group?
    2. Predict the weapon type, given the extent of damage
  - Use classifiers to determine if the weapons used to carry out attacks have changed over the years. Similarly, have the target sites changed over the years?

- **Functional Requirements**

  - Hardware /Software Requirements

- Hardware requirement

  1. Laptop or Desktop

- Software Requirement

  1. Spyder / Jupiter Notebook (ANACONDA)

Database Design

The Data Base We Required In This Project Is " GLOBAL TERRORISM DATA "

Application Work Flow

(This section displays the flow of information in the application)

We have not make any Application Software. We have done project on machine learning only with codes.

Screenshots

Q1. Perform analysis of the data and use Native Bayes and Nearest Neighbor models to predict the following:

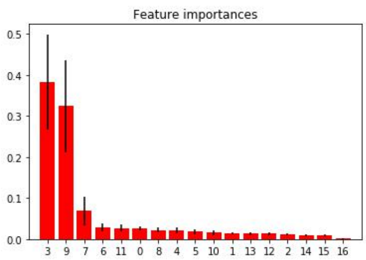
1.Predict the terrorist group, given other data fields. What are the attributes that best correlate to terrorist group?

At first we select all the features and get accuracy with 10 terrorist groups :

Naive Bayes :- 0.25
K-nearest neighbor :- 0.556

```
Confusion Matrix :
 [[952   1   0   0  76   0   0 178   0   0]
 [ 68   1   8   0   9   0   1 991   0   0]
 [112   0  10   0   0   0   0 478   0   0]
 [ 25   0  18   0   8   0   0 465   0   0]
 [ 79   0   0   0  11   0   0 189   0   0]
 [  6   0  14   0   0   0   0 240   0   0]
 [  0   0   1   0   0   0   0 230   0   0]
 [  0   0   4   0   0   0   0 214   0   0]
 [  0   0   0   0   0   0   0 184   0   0]
 [  6   0   5   0   0   0   0 160   1   0]]
Accuracy Score :         0.250368809273
Confusion Matrix :
 [[925  94  65  37  16  11  15  17  11  16]
 [114 748  57  58  29  15  21  12  12  12]
 [ 94  88 330  36  12  11  12   7   5   5]
 [ 78  87  43 236  14  20  17   6   4  11]
 [ 45  74  24  17  91   8  12   2   2   4]
 [ 41  31  32  24  19  90   9   3   3   8]
 [ 37  26  19  13   7   2 108  11   6   2]
 [ 39  53  14  12   4   5  11  61  17   2]
 [ 34  53  11  18   2   5  11  16  32   2]
 [ 63  39  16  17   6   7   2   1   1  20]]
Accuracy Score :         0.556585879874
```

selecting the features by random forest features selection we get a graph :



Choosing 3,9 and 7 means country_txt, target1, corp1 this features and removing the missing data we get the best accuracy with and without cross validation with 10 terrorist groups :

Naive Bayes :- 0.837
K-nearest neighbor :- 0.9888

```
Accuracy with cross validation ::        0.803359460909
Confusion Matrix :
 [[988   0   0   0   0   0   0   0   0   2]
 [  0 783   9   0   0   0  90   0   0   0]
 [  0  64 425   0   0   0   0   0   0   0]
 [ 31  13   0   0   0   0   0   0   0 312]
 [  0   0   0   0 207   0   0   0   0   0]
 [  0   1   2   0   0 222   0   0   0   0]
 [  0   0   0   0   0   0 171   0   0   0]
 [  0   0  28   0   0   0   2   0   1   0]
 [  0   0   2   0   0   0   1   0   1   0]
 [  3   5   2   0   0   0   0   0   0 126]]
Accuracy Score :         0.837295903753

Accuracy with cross validation ::        0.984683919348
Confusion Matrix :
 [[988   0   0   0   0   0   0   0   0   2]
 [  0 862   0   0   0   0  15   5   0   0]
 [  0   1 487   1   0   0   0   0   0   0]
 [  0   0   0 355   0   1   0   0   0   0]
 [  0   0   0   0 207   0   0   0   0   0]
 [  0   0   0   0   0 224   0   0   0   1]
 [  1   0   0   0   0   0 170   0   0   0]
 [  0   0   0   0   0   0   0  24   7   0]
 [  0   0   0   0   0   0   0   0   4   0]
 [  3   0   0   0   0   2   0   0   0 131]]
Accuracy Score :         0.988828415927
```

2.Predict the weapon type, given the extent of damage

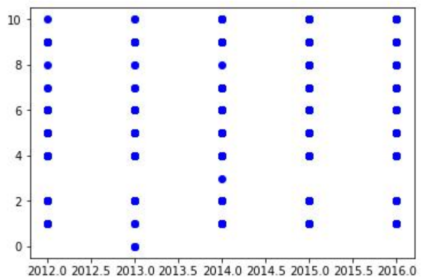
We get the accuracy with features nkill, nwound as human damages and propextent_txt as property damages :

Naive Bayes :- 0.533
K-nearest neighbor :-0.612

```
Confusion Matrix :
 [[   0    2    0    1    0    0    0]
 [   7  809  162  438    0    0   11]
 [   0  239  176  162    0    0    0]
 [   0   35   19  257    0    0    0]
 [   1    6    1    2    0    0    0]
 [   0    1    0    0    0    0    0]
 [   0    1    0    0    0    0    0]]
Accuracy Score :          0.5330472103
Confusion Matrix :
 [[    0     2     1     0     0     0     0]
 [    0  1257   168     2     0     0     0]
 [    0   408   168     1     0     0     0]
 [    0   296    14     1     0     0     0]
 [    0     7     3     0     0     0     0]
 [    0     1     0     0     0     0     0]
 [    0     1     0     0     0     0     0]]
Accuracy Score :          0.612017167382
```

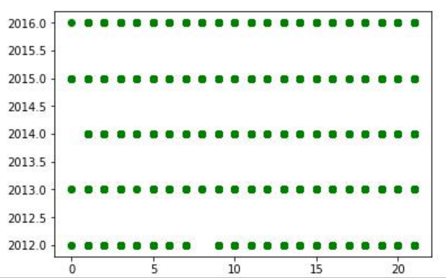
Q2. Use classifiers to determine if the weapons used to carry out attacks have changed over the years. Similarly, have the target sites changed over the years?

We get the bellow graph by maping year with weapon's type :



We can see from the above graph that almost same weapons are used from the starting of 2012 to 2016. From this we can conclude that there have no change in weapon type over the years

We get the bellow graph by maping year with weapon's type :



We can see from the above graph that targets places are almost same from the starting of 2012 to 2016. From this we can conclude that there have no change in Target places over the years

Future Scope of Improvements

Further work may be done by increasing the classes even further and seeing how it effects the predicted accuracy. Feature selection should be then run again on this new dataset. Also as the classes increase it is predicted that the accuracy will lower. This should give more false positives. It may also be useful to try adding weights to the classes, as more classes that are added may by be over shadowed by the larger classes.

Code

```
import os
import numpy as np
import pandas as pd
from sklearn import model_selection as ms
from sklearn import metrics as mt
from sklearn import preprocessing as pr
from sklearn import pipeline as pl
from sklearn import naive_bayes as nb
from sklearn import neighbors as ngb
import matplotlib.pyplot as plt



def printscore(prediction,actual):
    print("Confusion Matrix : \n", mt.confusion_matrix(actual,prediction))
    print("Accuracy Score : \t" , mt.accuracy_score(actual,prediction))



def pred_gname():

    os.chdir("G:/ML with Python/mldata/datafiles")
    gtd=pd.read_csv("terrsm1.csv",encoding = "iso-8859-1")

    gtd=gtd.drop("iyear",axis=1)
    gtd=gtd.drop("imonth",axis=1)
    gtd=gtd.drop("iday",axis=1)
    gtd=gtd.drop("city",axis=1)
    gtd=gtd.drop("attacktype1_txt",axis=1)
    gtd=gtd.drop("targtype1_txt",axis=1)
    gtd=gtd.drop("weaptype1_txt",axis=1)
    gtd=gtd.drop("weapsubtype1_txt",axis=1)
    gtd=gtd.drop("weapdetail",axis=1)
    gtd=gtd.drop("propextent_txt",axis=1)
    gtd=gtd.drop("ransomamt",axis=1)
    gtd=gtd.drop("nkill",axis=1)
    gtd=gtd.drop("nwound",axis=1)
    gtd=gtd.drop("target1",axis=1)
    gtd=gtd.drop("corp1",axis=1)



    test1 = gtd[gtd.gname == 'Taliban']

    test2 = gtd[gtd.gname == 'Islamic State of Iraq and the Levant (ISIL)']

    test3 = gtd[gtd.gname == 'Al-Shabaab']

    test4 = gtd[gtd.gname == 'Boko Haram']

    test5 = gtd[gtd.gname == 'Maoists']

    test6 = gtd[gtd.gname == "New People\'s Army (NPA)"]

    test7 = gtd[gtd.gname == 'Kurdistan Workers\' Party (PKK)']

    test8 = gtd[gtd.gname == 'Houthi extremists (Ansar Allah)']

    test9 = gtd[gtd.gname == 'Al-Qaida in the Arabian Peninsula (AQAP)']
```

```
test10 = gtd[gtd.gname == 'Tehrik-i-Taliban Pakistan (TTP)']


frames = [test1, test2, test3, test4, test5, test6, test7, test8, test9, test10]

result = pd.concat(frames)



leenc=pr.LabelEncoder()


result["natlty1_txt"]=leenc.fit_transform(result["natlty1_txt"])
result["natlty1_txt"]=result["natlty1_txt"]

result["country_txt"]=leenc.fit_transform(result["country_txt"])
result["country_txt"]=result["country_txt"]
'''
result["corp1"]=leenc.fit_transform(result["corp1"])
result["corp1"]=result["corp1"]
'''



print((result[["natlty1_txt"]]==65).sum())

result[["natlty1_txt"]]=result[["natlty1_txt"]].replace(65,np.NaN)

print((result[["natlty1_txt"]]==65).sum())

print(pd.value_counts(result["natlty1_txt"]))


'''
print((result[["corp1"]]==2909).sum())

result[["corp1"]]=result[["corp1"]].replace(2909,np.NaN)

print((result[["corp1"]]==2909).sum())

print(pd.value_counts(result["corp1"]))
'''

result.dropna(inplace=True)

result['organisation'] = result['gname'].map({'Taliban': 0,

                                'Islamic State of Iraq and the Levant (ISIL)': 1,

                                'Al-Shabaab': 2,

                                'Boko Haram': 3,

                                'Maoists': 4,

                                'New People\'s Army (NPA)': 5,

                                'Kurdistan Workers\' Party (PKK)': 6,

                                'Houthi extremists (Ansar Allah)': 7,

                                'Al-Qaida in the Arabian Peninsula (AQAP)': 8,
```

```
                              'Tehrik-i-Taliban Pakistan (TTP)': 9,

                              'Donetsk People\'s Republic': 10

                             }).astype(int)


result = result.drop(['gname'], axis=1)

target = result["organisation"]

data = result.drop(["organisation"], axis=1)



Xtrain, Xtest, ytrain, ytest =ms.train_test_split(data, target,test_size=0.2, random_state=42)


'''
forest = rfc(n_estimators=500,random_state=0)

forest.fit(Xtrain, ytrain)
importances = forest.feature_importances_
std = np.std([tree.feature_importances_ for tree in forest.estimators_],axis=0)
indices = np.argsort(importances)[::-1]

# Print the feature ranking
print("Feature ranking:")

for f in range(Xtrain.shape[1]):
print("%d. feature %d (%f)" % (f + 1, indices[f], importances[indices[f]]))

# Plot the feature importances of the forest
plt.figure()
plt.title("Feature importances")
plt.bar(range(Xtrain.shape[1]), importances[indices],
color="r", yerr=std[indices], align="center")
plt.xticks(range(Xtrain.shape[1]), indices)
plt.xlim([-1, Xtrain.shape[1]])
plt.show()
'''


model=nb.GaussianNB()
#nb.MultinomialNB()
model.fit(Xtrain,ytrain)

scores =ms.cross_val_score(model,Xtrain,ytrain,scoring="accuracy",cv=10)

print("\nAccuracy with cross validation :: \t", scores.mean())

predicted=model.predict(Xtest)

printscore(predicted,ytest)


knnclf=pl.Pipeline([("std",pr.StandardScaler()),
                ("clf",ngb.KNeighborsClassifier())
                ])
```

```
    knnclf=ngb.KNeighborsClassifier(n_neighbors=1,weights="uniform",algorithm="brute").fit(Xtrain,ytrain)

    scores =ms.cross_val_score(knnclf,Xtrain,ytrain,scoring="accuracy",cv=10)

    print("\nAccuracy with cross validation :: \t", scores.mean())


    predict=knnclf.predict(Xtest)

    printscore(predict,ytest)

    for actual,prediction in zip(ytest,predict):
        print("Actual :: {} , Prediction :: {}".format(actual,prediction))

def pred_weaptype():
    os.chdir("G:/ML with Python/mldata/datafiles")
    gtd=pd.read_csv("terrsm1.csv",encoding = "iso-8859-1")

    gtd=gtd.drop("iyear",axis=1)
    gtd=gtd.drop("imonth",axis=1)
    gtd=gtd.drop("iday",axis=1)
    gtd=gtd.drop("city",axis=1)
    gtd=gtd.drop("attacktype1_txt",axis=1)
    gtd=gtd.drop("targtype1_txt",axis=1)
    gtd=gtd.drop("weapsubtype1_txt",axis=1)
    gtd=gtd.drop("weapdetail",axis=1)
    gtd=gtd.drop("country_txt",axis=1)
    gtd=gtd.drop("corp1",axis=1)
    gtd=gtd.drop("natlty1_txt",axis=1)
    gtd=gtd.drop("ransomamt",axis=1)
    gtd=gtd.drop("target1",axis=1)
    gtd=gtd.drop("gname",axis=1)




    leenc=pr.LabelEncoder()

    gtd["propextent_txt"]

    gtd["weaptype1_txt"]=leenc.fit_transform(gtd["weaptype1_txt"])
    gtd["weaptype1_txt"]=gtd["weaptype1_txt"]
    gtd["weaptype1_txt"]


    print((gtd[["propextent_txt"]]=="Unknown").sum())

    gtd[["propextent_txt"]]=gtd[["propextent_txt"]].replace("Unknown",np.NaN)

    print((gtd[["propextent_txt"]]=="Unknown").sum())

    print(pd.value_counts(gtd["propextent_txt"]))



    print((gtd[["weaptype1_txt"]]==9).sum())

    gtd[["weaptype1_txt"]]=gtd[["weaptype1_txt"]].replace(9,np.NaN)
```

```
    print((gtd[["weaptype1_txt"]]==9).sum())

    print(pd.value_counts(gtd["weaptype1_txt"]))


    gtd.dropna(inplace=True)


    gtd[["propextent_txt"]]=(gtd["propextent_txt"]!="Minor").astype(np.int)



    target = gtd["weaptype1_txt"]

    data = gtd.drop(["weaptype1_txt"], axis=1)
    data




    Xtrain, Xtest, ytrain, ytest =ms.train_test_split(data, target,test_size=0.2, random_state=42)

    model=nb.MultinomialNB()
    model.fit(Xtrain,ytrain)

    predicted=model.predict(Xtest)

    printscore(predicted,ytest)





    knnclf=pl.Pipeline([("std",pr.StandardScaler()),
                ("clf",ngb.KNeighborsClassifier())
                ])


    knnclf=ngb.KNeighborsClassifier().fit(Xtrain,ytrain)
    predict=knnclf.predict(Xtest)

    printscore(predict,ytest)

    for actual,prediction in zip(ytest,predicted):
        print("Actual :: {} , Prediction :: {}".format(actual,prediction))



def det_change():
    os.chdir("G:/ML with Python/mldata/datafiles")
    gtd=pd.read_csv("terrsm1.csv",encoding = "iso-8859-1")

    gtd=gtd.drop("corp1",axis=1)
    gtd=gtd.drop("natlty1_txt",axis=1)
    gtd=gtd.drop("country_txt",axis=1)
    gtd=gtd.drop("imonth",axis=1)
    gtd=gtd.drop("iday",axis=1)
    gtd=gtd.drop("city",axis=1)
    gtd=gtd.drop("attacktype1_txt",axis=1)
    gtd=gtd.drop("weapsubtype1_txt",axis=1)
    gtd=gtd.drop("weapdetail",axis=1)
    gtd=gtd.drop("propextent_txt",axis=1)
    gtd=gtd.drop("ransomamt",axis=1)
    gtd=gtd.drop("nkill",axis=1)
```

```
    gtd=gtd.drop("nwound",axis=1)
    gtd=gtd.drop("target1",axis=1)

    gtd.head()
    leenc=pr.LabelEncoder()


    gtd["weaptype1_txt"]=leenc.fit_transform(gtd["weaptype1_txt"])
    gtd["targtype1_txt"]=leenc.fit_transform(gtd["targtype1_txt"])




    plt.plot(gtd["weaptype1_txt"],gtd["iyear"],'bo')
    plt.show()

    plt.plot(gtd["targtype1_txt"],gtd["iyear"],'go')
    plt.show()


def switch_case(arg):
    switchers={ 1: pred_gname, 2: pred_weaptype, 3: det_change }
    func = switchers.get(arg,lambda:"Invalid input")
    print(func())



print("1. Predicting Terrorist Group By Naive Bayes & K nearest neighbor \n\n2. Predicting Weapon's Types By Naive Bayes & K nearest neighbor \n\n3.Determine If There Have Any Change of Weapon's type & Target Places Over The Years? \n \n Enter Your Choice :: \t")
arg=int(input())
switch_case(arg)
```

Certificate

This is to certify that Mr/Ms [SOUMADIP MAJUMDAR] of [ASANSOL ENGINEERING COLLEGE], registration number: [161080120008], has successfully completed a project on [*GLOBAL TERRORISM DATA*] using [MACHINE LEARNING] under the guidance of Mr/Ms/Mrs [TITAS ROY CHOWDHURY].

--- ---

-

[TITAS ROY CHOWDHURY]

Globsyn Finishing School

Certificate

This is to certify that Mr/Ms [TUSHAR BANERJEE] of [ASANSOL ENGINEERING COLLEGE], registration number: [161080120009], has successfully completed a project on [*GLOBAL TERRORISM DATA*] using [MACHINE LEARNING] under the guidance of Mr/Ms/Mrs [TITAS ROY CHOWDHURY].

--- ---

-

[TITAS ROY CHOWDHURY]

Globsyn Finishing School

Certificate

This is to certify that Mr/Ms [ARPITA KARMAKAR] of [ASANSOL ENGINEERING COLLEGE], registration number: [161080120002], has successfully completed a project on [*GLOBAL TERRORISM DATA*] using [MACHINE LEARNING] under the guidance of Mr/Ms/Mrs [TITAS ROY CHOWDHURY].

--- --

-

[TITAS ROY CHOWDHURY]

Globsyn Finishing School

Certificate

This is to certify that Mr/Ms [ANGSHUMAN DEY] of [ASANSOL ENGINEERING COLLEGE], registration number: [161080120001], has successfully completed a project on [*GLOBAL TERRORISM DATA*] using [MACHINE LEARNING] under the guidance of Mr/Ms/Mrs [TITAS ROY CHOWDHURY].

--- ---

-

[TITAS ROY CHOWDHURY]

Globsyn Finishing School

Certificate

This is to certify that Mr/Ms [BINAY KUMAR BURNWAL] of [ASANSOL ENGINEERING COLLEGE], registration number: [161080120003], has successfully completed a project on [*GLOBAL TERRORISM DATA*] using [MACHINE LEARNING] under the guidance of Mr/Ms/Mrs [TITAS ROY CHOWDHURY].

--- --

-

[TITAS ROY CHOWDHURY]

Globsyn Finishing School

Certificate

This is to certify that Mr/Ms [VISHAL SHARMA] of [ASANSOL ENGINEERING COLLEGE], registration number: [151080110124], has successfully completed a project on [*GLOBAL TERRORISM DATA*] using [MACHINE LEARNING] under the guidance of Mr/Ms/Mrs [TITAS ROY CHOWDHURY].

--- ---

-

[TITAS ROY CHOWDHURY]

Globsyn Finishing School