

▼ To Identify Terrorist Events using Event Triggers

► Imports

↳ 10 cells hidden

► Global Objects

↳ 2 cells hidden

► Prepare data for classification

↳ 12 cells hidden

▼ Classification

```
Train_X = [' '.join(doc) for doc in train_file_data_list]
Test_X = [' '.join(doc) for doc in test_file_data_list]
Train_Y = train_labels
Test_Y = test_labels
corpus = Train_X + Test_X
```

```
Train_Y = Encoder.fit_transform(Train_Y)
Test_Y = Encoder.fit_transform(Test_Y)
```

```
Tfidf_vect = TfidfVectorizer()
Tfidf_vect.fit(corpus)
Train_X_Tfidf = Tfidf_vect.transform(Train_X)
Test_X_Tfidf = Tfidf_vect.transform(Test_X)
```

▼ Naive Bayes Classifier

```
Naive = naive_bayes.MultinomialNB()
Naive.fit(Train_X_Tfidf, Train_Y)
predictions_NB = Naive.predict(Test_X_Tfidf)
```

```
print(classification_report(predictions_NB, Test_Y, zero_division=0))
```

```

┌>

```

	precision	recall	f1-score	support
0	0.89	0.44	0.59	146
1	0.00	0.00	0.00	0
2	0.00	0.00	0.00	0
3	0.00	0.00	0.00	0
4	0.50	0.05	0.08	22
5	0.00	0.00	0.00	0
6	0.00	0.00	0.00	0
7	0.00	0.00	0.00	0
8	0.00	0.00	0.00	0
9	0.00	0.00	0.00	2
10	0.00	0.00	0.00	0
11	0.00	0.00	0.00	0
12	0.00	0.00	0.00	0
13	0.00	0.00	0.00	0
15	0.00	0.00	0.00	17
accuracy			0.35	187
macro avg	0.09	0.03	0.04	187
weighted avg	0.75	0.35	0.47	187

```
print(precision_score(predictions_NB, Test_Y, zero_division=0, average='weighted'))
print(recall_score(predictions_NB, Test_Y, zero_division=0, average='weighted'))
print(f1_score(predictions_NB, Test_Y, zero_division=0, average='weighted'))
```

```

┌> 0.7528223410576351
    0.34759358288770054
    0.4682251549493859
```

▼ SVM Classifier

```
SVM = svm.SVC(C=1.0, kernel='linear', degree=3, gamma='auto')
SVM.fit(Train_X_Tfidf,Train_Y)
predictions_SVM = SVM.predict(Test_X_Tfidf)
```

```
print(classification_report(predictions_SVM, Test_Y, zero_division=0))
```

```

precision    recall  f1-score   support

0           0.61      0.47      0.53         94
1           0.00      0.00      0.00          0
2           0.00      0.00      0.00          3
3           0.00      0.00      0.00          1
4           0.00      0.00      0.00         17
5           0.00      0.00      0.00          0
6           0.00      0.00      0.00          1
7           0.00      0.00      0.00          5
8           0.00      0.00      0.00          0
9           0.00      0.00      0.00          4
10          0.00      0.00      0.00          0
11          0.00      0.00      0.00          1
12          0.00      0.00      0.00          0
13          0.00      0.00      0.00          4
14          0.00      0.00      0.00         13
15          0.00      0.00      0.00         44

 accuracy          0.24         187
 macro avg          0.04         187
weighted avg          0.31         187
```

```
print(precision_score(predictions_SVM, Test_Y, zero_division=0, average='weighted'))
print(recall_score(predictions_SVM, Test_Y, zero_division=0, average='weighted'))
print(f1_score(predictions_SVM, Test_Y, zero_division=0, average='weighted'))
```

```

0.30718954248366015
0.23529411764705882
0.26647767540751244
```

```
SVM = svm.SVC(C=1.0, kernel='rbf', degree=3, gamma='auto')
SVM.fit(Train_X_Tfidf,Train_Y)
predictions_SVM = SVM.predict(Test_X_Tfidf)
```

```
print(classification_report(predictions_SVM, Test_Y, zero_division=0))
```

```

precision    recall  f1-score   support

0           1.00      0.39      0.56         187
1           0.00      0.00      0.00          0
2           0.00      0.00      0.00          0
3           0.00      0.00      0.00          0
4           0.00      0.00      0.00          0
5           0.00      0.00      0.00          0
6           0.00      0.00      0.00          0
7           0.00      0.00      0.00          0
8           0.00      0.00      0.00          0
9           0.00      0.00      0.00          0
10          0.00      0.00      0.00          0
11          0.00      0.00      0.00          0
12          0.00      0.00      0.00          0
13          0.00      0.00      0.00          0

 accuracy          0.39         187
 macro avg          0.07         187
weighted avg          1.00         187
```

```
print(precision_score(predictions_SVM, Test_Y, zero_division=0, average='weighted'))
print(recall_score(predictions_SVM, Test_Y, zero_division=0, average='weighted'))
print(f1_score(predictions_SVM, Test_Y, zero_division=0, average='weighted'))
```

```

1.0
0.3850267379679144
0.555984555984556
```

▼ Random Forest

```
RF = RandomForestClassifier()
RF.fit(Train_X_Tfidf,Train_Y)
predictions_RF = RF.predict(Test_X_Tfidf)
```

```
print(classification_report(predictions_RF, Test_Y, zero_division=0))
```

	precision	recall	f1-score	support
0	0.62	0.58	0.60	78
1	0.00	0.00	0.00	0
2	0.00	0.00	0.00	3
3	0.00	0.00	0.00	0
4	0.50	0.03	0.05	37
5	0.00	0.00	0.00	0
6	0.00	0.00	0.00	0
7	0.00	0.00	0.00	4
8	0.00	0.00	0.00	0
9	0.00	0.00	0.00	8
10	0.00	0.00	0.00	0
11	0.00	0.00	0.00	2
12	0.00	0.00	0.00	0
13	0.00	0.00	0.00	1
14	0.00	0.00	0.00	8
15	0.00	0.00	0.00	46
accuracy			0.25	187
macro avg	0.07	0.04	0.04	187
weighted avg	0.36	0.25	0.26	187

```
print(precision_score(predictions_RF, Test_Y, zero_division=0, average='weighted'))
print(recall_score(predictions_RF, Test_Y, zero_division=0, average='weighted'))
print(f1_score(predictions_RF, Test_Y, zero_division=0, average='weighted'))
```

```
0.35962566844919786
0.24598930481283418
0.26041409570821333
```

XGBoost Classifier

```
XGB = XGBClassifier()
XGB.fit(Train_X_Tfidf,Train_Y)
predictions_XGB = XGB.predict(Test_X_Tfidf)
```

```
print(classification_report(predictions_XGB, Test_Y, zero_division=0))
```

	precision	recall	f1-score	support
0	0.62	0.61	0.62	74
1	0.00	0.00	0.00	0
2	0.00	0.00	0.00	4
3	0.00	0.00	0.00	6
4	0.50	0.05	0.10	19
5	0.00	0.00	0.00	1
6	0.00	0.00	0.00	2
7	0.00	0.00	0.00	6
8	0.00	0.00	0.00	0
9	0.00	0.00	0.00	13
10	0.00	0.00	0.00	0
11	0.00	0.00	0.00	1
12	0.00	0.00	0.00	0
13	0.00	0.00	0.00	3
14	0.00	0.00	0.00	24
15	0.00	0.00	0.00	34
accuracy			0.25	187
macro avg	0.07	0.04	0.04	187
weighted avg	0.30	0.25	0.25	187

```
print(precision_score(predictions_XGB, Test_Y, zero_division=0, average='weighted'))
print(recall_score(predictions_XGB, Test_Y, zero_division=0, average='weighted'))
print(f1_score(predictions_XGB, Test_Y, zero_division=0, average='weighted'))
```

```
0.29812834224598933
0.24598930481283424
0.253614770939509
```

▼ Classification N-Gram (2,3)

```
Tfidf_vect = TfidfVectorizer(ngram_range=(2,3))
Tfidf_vect.fit(corpus)
Train_X_Tfidf = Tfidf_vect.transform(Train_X)
Test_X_Tfidf = Tfidf_vect.transform(Test_X)
```

▼ Naive Bayes Classifier

```
Naive = naive_bayes.MultinomialNB()
Naive.fit(Train_X_Tfidf,Train_Y)
predictions_NB = Naive.predict(Test_X_Tfidf)
```

```
print(classification_report(predictions_NB, Test_Y, zero_division=0))
```

📄		precision	recall	f1-score	support
	0	0.96	0.41	0.57	169
	1	0.00	0.00	0.00	0
	2	0.00	0.00	0.00	0
	3	0.00	0.00	0.00	0
	4	0.00	0.00	0.00	12
	5	0.00	0.00	0.00	0
	6	0.00	0.00	0.00	0
	7	0.00	0.00	0.00	0
	8	0.00	0.00	0.00	0
	9	0.00	0.00	0.00	0
	10	0.00	0.00	0.00	0
	11	0.00	0.00	0.00	0
	12	0.00	0.00	0.00	0
	13	0.00	0.00	0.00	0
	14	0.00	0.00	0.00	1
	15	0.00	0.00	0.00	5
	accuracy			0.37	187
	macro avg	0.06	0.03	0.04	187
	weighted avg	0.87	0.37	0.52	187

```
print(precision_score(predictions_NB, Test_Y, zero_division=0, average='weighted'))
print(recall_score(predictions_NB, Test_Y, zero_division=0, average='weighted'))
print(f1_score(predictions_NB, Test_Y, zero_division=0, average='weighted'))
```

```
📄 0.8660873440285205
0.3689839572192513
0.517496172365589
```

▼ SVM Classifier

```
SVM = svm.SVC(C=1.0, kernel='linear', degree=3, gamma='auto')
SVM.fit(Train_X_Tfidf,Train_Y)
predictions_SVM = SVM.predict(Test_X_Tfidf)
```

```
print(classification_report(predictions_SVM, Test_Y, zero_division=0))
```

```
📄
```


	precision	recall	f1-score	support
0	0.90	0.41	0.57	158
1	0.00	0.00	0.00	0
2	0.00	0.00	0.00	0
3	0.00	0.00	0.00	0
4	0.00	0.00	0.00	13
5	0.00	0.00	0.00	0
6	0.00	0.00	0.00	0
7	0.00	0.00	0.00	0
8	0.00	0.00	0.00	0
9	0.00	0.00	0.00	1
10	0.00	0.00	0.00	0
11	0.00	0.00	0.00	0
12	0.00	0.00	0.00	0
13	0.00	0.00	0.00	0
14	0.00	0.00	0.00	1
15	0.00	0.00	0.00	14
accuracy			0.35	187
macro avg	0.06	0.03	0.04	187
weighted avg	0.76	0.35	0.48	187

```
print(precision_score(predictions_SVM, Test_Y, zero_division=0, average='weighted'))
print(recall_score(predictions_SVM, Test_Y, zero_division=0, average='weighted'))
print(f1_score(predictions_SVM, Test_Y, zero_division=0, average='weighted'))
```

```
📄 0.7627748068924539
0.34759358288770054
0.4775633573587538
```

```
SVM = svm.SVC(C=1.0, kernel='rbf', degree=3, gamma='auto')
SVM.fit(Train_X_Tfidf,Train_Y)
predictions_SVM = SVM.predict(Test_X_Tfidf)
```

```
print(classification_report(predictions_SVM, Test_Y, zero_division=0))
```

		precision	recall	f1-score	support
	0	1.00	0.39	0.56	187
	1	0.00	0.00	0.00	0
	2	0.00	0.00	0.00	0
	3	0.00	0.00	0.00	0
	4	0.00	0.00	0.00	0
	5	0.00	0.00	0.00	0
	6	0.00	0.00	0.00	0
	7	0.00	0.00	0.00	0
	8	0.00	0.00	0.00	0
	9	0.00	0.00	0.00	0
	10	0.00	0.00	0.00	0
	11	0.00	0.00	0.00	0
	12	0.00	0.00	0.00	0
	13	0.00	0.00	0.00	0
	accuracy			0.39	187
	macro avg	0.07	0.03	0.04	187
	weighted avg	1.00	0.39	0.56	187

```
print(precision_score(predictions_SVM, Test_Y, zero_division=0, average='weighted'))
print(recall_score(predictions_SVM, Test_Y, zero_division=0, average='weighted'))
print(f1_score(predictions_SVM, Test_Y, zero_division=0, average='weighted'))
```

```
📄 1.0
0.3850267379679144
0.555984555984556
```

Random Forest

```
RF = RandomForestClassifier()
RF.fit(Train_X_Tfidf,Train_Y)
predictions_RF = RF.predict(Test_X_Tfidf)
```

```
print(classification_report(predictions_RF, Test_Y, zero_division=0))
```

```
📄
```

	precision	recall	f1-score	support
0	0.90	0.41	0.57	157
1	0.00	0.00	0.00	0
2	0.00	0.00	0.00	0
3	0.00	0.00	0.00	0
4	0.00	0.00	0.00	15
5	0.00	0.00	0.00	0
6	0.00	0.00	0.00	0
7	0.00	0.00	0.00	0
8	0.00	0.00	0.00	0
9	0.00	0.00	0.00	3
10	0.00	0.00	0.00	0
11	0.00	0.00	0.00	0
12	0.00	0.00	0.00	0
13	0.00	0.00	0.00	0
14	0.00	0.00	0.00	3
15	0.00	0.00	0.00	9
accuracy				0.35
macro avg				0.06
weighted avg				0.35

```
print(precision_score(predictions_RF, Test_Y, zero_division=0, average='weighted'))
print(recall_score(predictions_RF, Test_Y, zero_division=0, average='weighted'))
print(f1_score(predictions_RF, Test_Y, zero_division=0, average='weighted'))
```

```
0.757947118241236
0.34759358288770054
0.47661303505125746
```

XGBoost Classifier

```
XGB = XGBClassifier()
XGB.fit(Train_X_Tfidf,Train_Y)
predictions_XGB = XGB.predict(Test_X_Tfidf)
```

```
print(classification_report(predictions_XGB, Test_Y, zero_division=0))
```

	precision	recall	f1-score	support
0	0.62	0.46	0.53	97
1	0.00	0.00	0.00	0
2	0.00	0.00	0.00	3
3	0.00	0.00	0.00	3
4	0.00	0.00	0.00	17
5	0.00	0.00	0.00	0
6	0.00	0.00	0.00	2
7	0.00	0.00	0.00	0
8	0.00	0.00	0.00	0
9	0.00	0.00	0.00	5
10	0.00	0.00	0.00	0
11	0.00	0.00	0.00	1
12	0.00	0.00	0.00	0
13	0.00	0.00	0.00	2
14	0.00	0.00	0.00	18
15	0.00	0.00	0.00	39
accuracy				0.24
macro avg				0.04
weighted avg				0.32

```
print(precision_score(predictions_XGB, Test_Y, zero_division=0, average='weighted'))
print(recall_score(predictions_XGB, Test_Y, zero_division=0, average='weighted'))
print(f1_score(predictions_XGB, Test_Y, zero_division=0, average='weighted'))
```

```
0.32419786096256686
0.24064171122994651
0.2762395975065658
```

Classification N-gram (3,4)

```
Tfidf_vect = TfidfVectorizer(ngram_range=(3,4))
Tfidf_vect.fit(corpus)
Train_X_Tfidf = Tfidf_vect.transform(Train_X)
```

```
Test_X_Tfidf = Tfidf_vect.transform(Test_X)
```

▼ Naive Bayes Classifier

```
Naive = naive_bayes.MultinomialNB()
Naive.fit(Train_X_Tfidf, Train_Y)
predictions_NB = Naive.predict(Test_X_Tfidf)
```

```
print(classification_report(predictions_NB, Test_Y, zero_division=0))
```

```

precision    recall  f1-score   support

0           0.99      0.39      0.55       184
1           0.00      0.00      0.00         0
2           0.00      0.00      0.00         0
3           0.00      0.00      0.00         0
4           0.00      0.00      0.00         1
5           0.00      0.00      0.00         0
6           0.00      0.00      0.00         0
7           0.00      0.00      0.00         0
8           0.00      0.00      0.00         0
9           0.00      0.00      0.00         0
10          0.00      0.00      0.00         0
11          0.00      0.00      0.00         0
12          0.00      0.00      0.00         0
13          0.00      0.00      0.00         0
14          0.00      0.00      0.00         1
15          0.00      0.00      0.00         1

 accuracy          0.38       187
 macro avg         0.06       187
 weighted avg      0.97       187
```

```
print(precision_score(predictions_NB, Test_Y, zero_division=0, average='weighted'))
print(recall_score(predictions_NB, Test_Y, zero_division=0, average='weighted'))
print(f1_score(predictions_NB, Test_Y, zero_division=0, average='weighted'))
```

```

0.9702911467617351
0.37967914438502676
0.5457887700534759
```

▼ SVM Classifier

```
SVM = svm.SVC(C=1.0, kernel='linear', degree=3, gamma='auto')
SVM.fit(Train_X_Tfidf, Train_Y)
predictions_SVM = SVM.predict(Test_X_Tfidf)
```

```
print(classification_report(predictions_SVM, Test_Y, zero_division=0))
```

```

precision    recall  f1-score   support

0           0.99      0.39      0.56       182
1           0.00      0.00      0.00         0
2           0.00      0.00      0.00         0
3           0.00      0.00      0.00         0
4           0.00      0.00      0.00         2
5           0.00      0.00      0.00         0
6           0.00      0.00      0.00         0
7           0.00      0.00      0.00         0
8           0.00      0.00      0.00         0
9           0.00      0.00      0.00         0
10          0.00      0.00      0.00         0
11          0.00      0.00      0.00         0
12          0.00      0.00      0.00         0
13          0.00      0.00      0.00         0
14          0.00      0.00      0.00         1
15          0.00      0.00      0.00         2

 accuracy          0.38       187
 macro avg         0.06       187
 weighted avg      0.96       187
```

```
print(precision_score(predictions_SVM, Test_Y, zero_division=0, average='weighted'))
print(recall_score(predictions_SVM, Test_Y, zero_division=0, average='weighted'))
print(f1_score(predictions_SVM, Test_Y, zero_division=0, average='weighted'))
```

```
0.9597445038621509
0.37967914438502676
0.5441071202998021
```

```
SVM = svm.SVC(C=1.0, kernel='rbf', degree=3, gamma='auto')
SVM.fit(Train_X_Tfidf,Train_Y)
predictions_SVM = SVM.predict(Test_X_Tfidf)
```

```
print(classification_report(predictions_SVM, Test_Y, zero_division=0))
```

	precision	recall	f1-score	support
0	1.00	0.39	0.56	187
1	0.00	0.00	0.00	0
2	0.00	0.00	0.00	0
3	0.00	0.00	0.00	0
4	0.00	0.00	0.00	0
5	0.00	0.00	0.00	0
6	0.00	0.00	0.00	0
7	0.00	0.00	0.00	0
8	0.00	0.00	0.00	0
9	0.00	0.00	0.00	0
10	0.00	0.00	0.00	0
11	0.00	0.00	0.00	0
12	0.00	0.00	0.00	0
13	0.00	0.00	0.00	0
accuracy			0.39	187
macro avg	0.07	0.03	0.04	187
weighted avg	1.00	0.39	0.56	187

```
print(precision_score(predictions_SVM, Test_Y, zero_division=0, average='weighted'))
print(recall_score(predictions_SVM, Test_Y, zero_division=0, average='weighted'))
print(f1_score(predictions_SVM, Test_Y, zero_division=0, average='weighted'))
```

```
1.0
0.3850267379679144
0.555984555984556
```