MSDS692_CGREEN_Final Project ML Code

March 4, 2022

1 CGreen - MSDS692 Project Code - ML and Extremist Ideology

This code takes in extremist attack information from 1970 to 2019 and tests whether KNN can accurately assign ideologicla motivation based on the characteristics of the attack.

1.1 Loading Packages and Setting Pandas Options

[1]:
```
# pandas
import pandas as pd

# sklearn
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn import model_selection
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso
from sklearn.linear_model import ElasticNet
from sklearn.svm import SVR
from sklearn.pipeline import Pipeline
from sklearn.metrics import mean_squared_error
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from matplotlib import pyplot
from sklearn.metrics import accuracy_score
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import RandomForestClassifier
```

```
from sklearn.ensemble import ExtraTreesClassifier

# plotting
!conda install -c conda-forge scikit-plot -y
from scikitplot.estimators import plot_feature_importances
import seaborn as sns
from matplotlib import pyplot
import matplotlib.pyplot as plt
from sklearn.metrics import plot_confusion_matrix
%matplotlib inline
sns.set()

# others
import numpy as np
import scipy.stats as stats
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.
```

[2]:
```
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.width', None)
pd.set_option('display.max_colwidth', None)
```

1.2 Importing Initial Datasets

Data for this project came in two parts. The GTD data contained attack information from 1970-2019 without ideological motivation. The PPT data contained information on specific groups and their ideologies. Merging the two allowed me to link together the attacks to dominant ideologies.

[3]:
```
df_GTD = pd.read_csv('20220210_GTD.csv')
df_PPT = pd.read_csv('20220210_PPT.csv')
```

[4]:
```
df_GTD.shape
```

[4]:
```
(3004, 35)
```

[5]:
```
df_PPT.shape
```

[5]:
```
(239, 45)
```

For the join operation, I reviewed the 2022 article by Lee and the 2021 article by Stratis for reminders on the merge function in Pandas.

[6]:
```
df_combined = pd.merge(df_GTD, df_PPT, on='ORGNAME', how='inner')
```

[7]:
```
df_combined.shape
```

[7]:
```
(2969, 79)
```

[8]:
```
df_combined.head()
```

[8]:
```
      Event ID#  Year  Month  Day       State           City        LAT  \
0  197001010002  1970      1    1    Illinois          Cairo  37.005105
1  197001120001  1970      1   12    New York  New York City  40.697132
2  197001130001  1970      1   13  Washington        Seattle  47.610786
3  197001140001  1970      1   14    Illinois      Champaign  40.116748
4  197001190002  1970      1   19  Washington        Seattle  47.610786

         LONG  \
0  -89.176269
1  -73.931351
2 -122.331306
3  -88.239270
4 -122.331306

                  Summary  \
0  1/1/1970: Unknown African American assailants fired several bullets at police
headquarters in Cairo, Illinois, United States. There were no casualties,
however, one bullet narrowly missed several police officers. This attack took
place during heightened racial tensions, including a Black boycott of White-
owned businesses, in Cairo Illinois.
1                1/12/1970: Unknown perpetrators threw a pipe bomb into the
vacant dean's office of James Madison High School in Brooklyn, New York, United
States. There were no casualties but the explosion caused minor damages. Earlier
in the day anti-war and pro-Black Panther statements were discovered painted
outside the walls of the school.
2                                             1/13/1970: Unknown perpetrators
firebombed Fuson's Department Store in Seattle, Washington, United States. There
were no casualties but the store sustained an estimated $17,000 in damages. Less
than a week earlier, the store owner shot and killed an African American male
attempting to rob the store.
3
1/14/1970: Suspected Black militants threw two firebombs into the Champaign
Police Department in Champaign, Illinois, United States. The building was
damaged and one police officer was severely burned.
4
1/17/1970: Three African Americans were suspected of detonating a bomb on the
Seattle University campus in Seattle, Washington, United States. There were no
casualties but the Liberal Arts and Garrand buildings sustained $2,200 in
damages.

   Successful  Suicide  Attack Type                Attack Type Text  \
```

```
0          1        0            2                 Armed Assault
1          1        0            3             Bombing/Explosion
2          1        0            7  Facility/Infrastructure Attack
3          1        0            7  Facility/Infrastructure Attack
4          1        0            3             Bombing/Explosion

   Target Type         Target Type Text  \
0            3                   Police
1            8  Educational Institution
2            1                 Business
3            3                   Police
4            8  Educational Institution

                                                 Target             ORGNAME  \
0                              Cairo Police Headquarters  Black Nationalists
1                              James Madison High School  Black Nationalists
2           Fuson's Department Store, Seattle Washington  Black Nationalists
3                            Champaign Police Department  Black Nationalists
4  Liberal Arts and Garrand buildings, Seattle University  Black Nationalists

                                                         Motive  \
0
To protest the Cairo Illinois Police Deparment
1  Suspected motives were to protest the Vietnam War and/or show support for the
Black Panther Party and/or show support for the Young Lords.
2                        Retaliation for the store owner who shot and killed an
African American attempting to commit a robbery at his store.
3
NaN
4                                              The incident took place during
disturbances between the Black Student Union and the university.

   Unaffiliated Individual  Claim  Weapon Type Weapon Type Text  \
0                        0    0.0            5         Firearms
1                        0    0.0            6       Explosives
2                        0    0.0            8       Incendiary
3                        0    0.0            8       Incendiary
4                        0    0.0            6       Explosives

   Weapon Sub-Type          Weapon Sub-Type Text   Number Killed  \
0              5.0              Unknown Gun Type             0.0
1             31.0                     Pipe Bomb             0.0
2             19.0  Molotov Cocktail/Petrol Bomb             0.0
3             19.0  Molotov Cocktail/Petrol Bomb             0.0
4             16.0        Unknown Explosive Type             0.0

   Number Wounded  Number Attackers Killed  Number Atackers Wounded  \
```

```
0            0.0                        0.0                        0.0
1            0.0                        0.0                        0.0
2            0.0                        0.0                        0.0
3            1.0                        0.0                        0.0
4            0.0                        0.0                        0.0

   Property Damage  Property Damage Extent          Damage Extent Text  \
0                1                     3.0  Minor (likely < $1 million)
1                1                     3.0  Minor (likely < $1 million)
2                1                     3.0  Minor (likely < $1 million)
3                1                     3.0  Minor (likely < $1 million)
4                1                     3.0  Minor (likely < $1 million)

   Kidnapping  Number Hostages  Ransom Demand  Hostage/Kidnapping Outcome  \
0         0.0              NaN            0.0                         NaN
1         0.0              NaN            0.0                         NaN
2         0.0              NaN            0.0                         NaN
3         0.0              NaN            0.0                         NaN
4         0.0              NaN            0.0                         NaN

   DOM_I  I_ETHNO  I_REL  I_REL_1  I_REL_2  I_REL_3  I_REL_4  I_REL_5  \
0      2        0      0        0        0        0        0        0
1      2        0      0        0        0        0        0        0
2      2        0      0        0        0        0        0        0
3      2        0      0        0        0        0        0        0
4      2        0      0        0        0        0        0        0

   I_REL_6  I_REL_7  I_REL_8  I_REL_9  I_REL_10  I_RACE  I_RACE_1  I_RACE_2  \
0        0        0        0        0         0       0         0         0
1        0        0        0        0         0       0         0         0
2        0        0        0        0         0       0         0         0
3        0        0        0        0         0       0         0         0
4        0        0        0        0         0       0         0         0

   I_RACE_3  I_LEFT  I_LEFT_1  I_LEFT_2  I_LEFT_3  I_LEFT_4  I_LEFT_5  \
0       0.0       1         0         0       0.0         0         1
1       0.0       1         0         0       0.0         0         1
2       0.0       1         0         0       0.0         0         1
3       0.0       1         0         0       0.0         0         1
4       0.0       1         0         0       0.0         0         1

   I_LEFT_6  I_RIGHT  I_RIGHT_1  I_RIGHT_2  I_RIGHT_3  I_RIGHT_4  I_SI  \
0         0        0          0          0          0          0     0
1         0        0          0          0          0          0     0
2         0        0          0          0          0          0     0
3         0        0          0          0          0          0     0
4         0        0          0          0          0          0     0
```

```
   I_SI_1  I_SI_2  I_SI_3  I_SI_4  I_SI_5  I_SI_6  I_SI_7  I_SI_8  I_SI_9  \
0       0       0       0       0       0       0     0.0       0       0
1       0       0       0       0       0       0     0.0       0       0
2       0       0       0       0       0       0     0.0       0       0
3       0       0       0       0       0       0     0.0       0       0
4       0       0       0       0       0       0     0.0       0       0

   I_SI_10  I_SI_11  I_SI_12  I_SI_13  I_SI_14
0        0        0        0        0        0
1        0        0        0        0        0
2        0        0        0        0        0
3        0        0        0        0        0
4        0        0        0        0        0
```

[9]:
```
df_combined.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2969 entries, 0 to 2968
Data columns (total 79 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   Event ID#                    2969 non-null   int64
 1   Year                         2969 non-null   int64
 2   Month                        2969 non-null   int64
 3   Day                          2969 non-null   int64
 4   State                        2969 non-null   object
 5   City                         2969 non-null   object
 6   LAT                          2968 non-null   float64
 7   LONG                         2968 non-null   float64
 8   Summary                      1927 non-null   object
 9   Successful                   2969 non-null   int64
 10  Suicide                      2969 non-null   int64
 11  Attack Type                  2969 non-null   int64
 12  Attack Type Text             2969 non-null   object
 13  Target Type                  2969 non-null   int64
 14  Target Type Text             2969 non-null   object
 15  Target                       2924 non-null   object
 16  ORGNAME                      2969 non-null   object
 17  Motive                       1577 non-null   object
 18  Unaffiliated Individual      2969 non-null   int64
 19  Claim                        1935 non-null   float64
 20  Weapon Type                  2969 non-null   int64
 21  Weapon Type Text             2969 non-null   object
 22  Weapon Sub-Type              2697 non-null   float64
 23  Weapon Sub-Type Text         2697 non-null   object
 24   Number Killed               2900 non-null   float64
```

25	Number Wounded	2879 non-null	float64
26	Number Attackers Killed	1978 non-null	float64
27	Number Atackers Wounded	1964 non-null	float64
28	Property Damage	2969 non-null	int64
29	Property Damage Extent	1703 non-null	float64
30	Damage Extent Text	1703 non-null	object
31	Kidnapping	2810 non-null	float64
32	Number Hostages	68 non-null	float64
33	Ransom Demand	2155 non-null	float64
34	Hostage/Kidnapping Outcome	47 non-null	float64
35	DOM_I	2969 non-null	int64
36	I_ETHNO	2969 non-null	int64
37	I_REL	2969 non-null	int64
38	I_REL_1	2969 non-null	int64
39	I_REL_2	2969 non-null	int64
40	I_REL_3	2969 non-null	int64
41	I_REL_4	2969 non-null	int64
42	I_REL_5	2969 non-null	int64
43	I_REL_6	2969 non-null	int64
44	I_REL_7	2969 non-null	int64
45	I_REL_8	2969 non-null	int64
46	I_REL_9	2969 non-null	int64
47	I_REL_10	2969 non-null	int64
48	I_RACE	2969 non-null	int64
49	I_RACE_1	2969 non-null	int64
50	I_RACE_2	2969 non-null	int64
51	I_RACE_3	2953 non-null	float64
52	I_LEFT	2969 non-null	int64
53	I_LEFT_1	2969 non-null	int64
54	I_LEFT_2	2969 non-null	int64
55	I_LEFT_3	2966 non-null	float64
56	I_LEFT_4	2969 non-null	int64
57	I_LEFT_5	2969 non-null	int64
58	I_LEFT_6	2969 non-null	int64
59	I_RIGHT	2969 non-null	int64
60	I_RIGHT_1	2969 non-null	int64
61	I_RIGHT_2	2969 non-null	int64
62	I_RIGHT_3	2969 non-null	int64
63	I_RIGHT_4	2969 non-null	int64
64	I_SI	2969 non-null	int64
65	I_SI_1	2969 non-null	int64
66	I_SI_2	2969 non-null	int64
67	I_SI_3	2969 non-null	int64
68	I_SI_4	2969 non-null	int64
69	I_SI_5	2969 non-null	int64
70	I_SI_6	2969 non-null	int64
71	I_SI_7	2967 non-null	float64
72	I_SI_8	2969 non-null	int64

```
 73   I_SI_9                      2969 non-null   int64
 74   I_SI_10                     2969 non-null   int64
 75   I_SI_11                     2969 non-null   int64
 76   I_SI_12                     2969 non-null   int64
 77   I_SI_13                     2969 non-null   int64
 78   I_SI_14                     2969 non-null   int64
dtypes: float64(16), int64(52), object(11)
memory usage: 1.8+ MB
```

Exporting the combined set for further analysis At this point I exported the combined set out to Excel for further analysis. It was easier to visualize the connections and features outside of Jupyter and it also gave me an expanded platform for data visualization.

[10]:
```
df_combined.to_excel("combined.xlsx")
```

1.3 Importing the Cleaned, Combined Dataset

After cleaning and shaping the data in Excel, I re-imported the dataset for use in our ML test.

[11]:
```
df = pd.read_csv('20220217 Combined for ML_No I Codes.csv')
```

[12]:
```
df.shape
```

[12]:
```
(2394, 19)
```

[13]:
```
df.head()
```

[13]:
```
       Event ID#  Year  Month  Day      State     City  \
0  197001010002  1970      1    1   Illinois    Cairo
1  197001020003  1970      1    2  Wisconsin  Madison
2  197001030001  1970      1    3  Wisconsin  Madison
3  197001050001  1970      1    1  Wisconsin  Baraboo
4  197001060001  1970      1    6   Colorado   Denver

                          Group  Unaffiliated Individual  Claim  \
0            Black Nationalists                        0      0
1               New Year's Gang                        0      1
2               New Year's Gang                        0      0
3  Weather Underground, Weathermen                     0      0
4           Left-Wing Militants                        0      0

   Successful  Suicide                            Type                Target  \
0           1        0                  Armed Assault                Police
1           1        0  Facility/Infrastructure Attack              Military
2           1        0  Facility/Infrastructure Attack  Government (General)
3           0        0               Bombing/Explosion              Military
4           1        0  Facility/Infrastructure Attack              Military
```

```
       Weapon  Casualties  Property Damage  Kidnapping  Ransom Demand  \
0    Firearms           0                1           0              0
1  Incendiary           0                1           0              0
2  Incendiary           0                1           0              0
3  Explosives           0                0           0              0
4  Incendiary           0                1           0              0

   Dominant Ideology
0  Extreme Left Wing
1       Single Issue
2       Single Issue
3  Extreme Left Wing
4  Extreme Left Wing
```

[14]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2394 entries, 0 to 2393
Data columns (total 19 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Event ID#               2394 non-null   int64
 1   Year                    2394 non-null   int64
 2   Month                   2394 non-null   int64
 3   Day                     2394 non-null   int64
 4   State                   2394 non-null   object
 5   City                    2394 non-null   object
 6   Group                   2394 non-null   object
 7   Unaffiliated Individual 2394 non-null   int64
 8   Claim                   2394 non-null   int64
 9   Successful              2394 non-null   int64
 10  Suicide                 2394 non-null   int64
 11  Type                    2394 non-null   object
 12  Target                  2394 non-null   object
 13  Weapon                  2394 non-null   object
 14  Casualties              2394 non-null   int64
 15  Property Damage         2394 non-null   int64
 16  Kidnapping              2394 non-null   int64
 17  Ransom Demand           2394 non-null   int64
 18  Dominant Ideology       2394 non-null   object
dtypes: int64(12), object(7)
memory usage: 355.5+ KB
```

1.4 Prepping the features for ML Test

I had to convert a number of features from categories/objects to numerical variables so that the algorithm could recognize them. I used my notes from MSDS650 and reviewed the 2021 article by Chen and the 2017 article by Moffitt for additional reference on this process.

[15]:
```
df = pd.get_dummies(df, columns=["Type", "Target", "Weapon", "Dominant␣
↪Ideology"])
```

[16]:
```
df.head()
```

[16]:
```
      Event ID#  Year  Month  Day      State     City  \
0  197001010002  1970      1    1   Illinois    Cairo
1  197001020003  1970      1    2  Wisconsin  Madison
2  197001030001  1970      1    3  Wisconsin  Madison
3  197001050001  1970      1    1  Wisconsin  Baraboo
4  197001060001  1970      1    6   Colorado   Denver

                               Group  Unaffiliated Individual  Claim  \
0                Black Nationalists                        0      0
1                   New Year's Gang                        0      1
2                   New Year's Gang                        0      0
3  Weather Underground, Weathermen                        0      0
4                Left-Wing Militants                       0      0

   Successful  Suicide  Casualties  Property Damage  Kidnapping  \
0           1        0           0                1           0
1           1        0           0                1           0
2           1        0           0                1           0
3           0        0           0                0           0
4           1        0           0                1           0

   Ransom Demand  Type_Armed Assault  Type_Assassination  \
0              0                   1                   0
1              0                   0                   0
2              0                   0                   0
3              0                   0                   0
4              0                   0                   0

   Type_Bombing/Explosion  Type_Facility/Infrastructure Attack  \
0                       0                                    0
1                       0                                    1
2                       0                                    1
3                       1                                    0
4                       0                                    1

   Type_Hijacking  Type_Hostage Taking (Barricade Incident)  \
0               0                                         0
1               0                                         0
2               0                                         0
3               0                                         0
4               0                                         0
```

```
   Type_Hostage Taking (Kidnapping)  Type_Unarmed Assault  Type_Unknown  \
0                                 0                     0             0
1                                 0                     0             0
2                                 0                     0             0
3                                 0                     0             0
4                                 0                     0             0

   Target_Abortion Related  Target_Airports & Aircraft  Target_Business  \
0                        0                           0                0
1                        0                           0                0
2                        0                           0                0
3                        0                           0                0
4                        0                           0                0

   Target_Educational Institution  Target_Food or Water Supply  \
0                               0                            0
1                               0                            0
2                               0                            0
3                               0                            0
4                               0                            0

   Target_Government (Diplomatic)  Target_Government (General)  \
0                               0                            0
1                               0                            0
2                               0                            1
3                               0                            0
4                               0                            0

   Target_Journalists & Media  Target_Maritime  Target_Military  Target_NGO  \
0                           0                0                0           0
1                           0                0                1           0
2                           0                0                0           0
3                           0                0                1           0
4                           0                0                1           0

   Target_Other  Target_Police  Target_Private Citizens & Property  \
0             0              1                                   0
1             0              0                                   0
2             0              0                                   0
3             0              0                                   0
4             0              0                                   0

   Target_Religious Figures/Institutions  Target_Telecommunication  \
0                                      0                         0
1                                      0                         0
2                                      0                         0
3                                      0                         0
```

```
4                                  0                         0

   Target_Terrorists/Non-State Militia  Target_Tourists  \
0                                    0                0
1                                    0                0
2                                    0                0
3                                    0                0
4                                    0                0

   Target_Transportation  Target_Unknown  Target_Utilities  \
0                      0               0                 0
1                      0               0                 0
2                      0               0                 0
3                      0               0                 0
4                      0               0                 0

   Target_Violent Political Party  Weapon_Biological  Weapon_Chemical  \
0                               0                  0                0
1                               0                  0                0
2                               0                  0                0
3                               0                  0                0
4                               0                  0                0

   Weapon_Explosives  Weapon_Fake Weapons  Weapon_Firearms  Weapon_Incendiary  \
0                  0                    0                1                  0
1                  0                    0                0                  1
2                  0                    0                0                  1
3                  1                    0                0                  0
4                  0                    0                0                  1

   Weapon_Melee  Weapon_Other  Weapon_Sabotage Equipment  Weapon_Unknown  \
0             0             0                          0               0
1             0             0                          0               0
2             0             0                          0               0
3             0             0                          0               0
4             0             0                          0               0

   Weapon_Vehicle (not to include vehicle-borne explosives, i.e., car or truck
bombs)  \
0
0
1
0
2
0
3
0
```

```
4
0

   Dominant Ideology_Ethnonationalist-Separatist  \
0                                              0
1                                              0
2                                              0
3                                              0
4                                              0

   Dominant Ideology_Extreme Left Wing  Dominant Ideology_Extreme Right Wing  \
0                                    1                                     0
1                                    0                                     0
2                                    0                                     0
3                                    1                                     0
4                                    1                                     0

   Dominant Ideology_Religious  Dominant Ideology_Single Issue  \
0                            0                               0
1                            0                               1
2                            0                               1
3                            0                               0
4                            0                               0

   Dominant Ideology_Unknown Ideology
0                                   0
1                                   0
2                                   0
3                                   0
4                                   0
```

1.5 Running the Data though the KNN classifier

There were five Dominant Ideologies (Ethnonationalist-Separatist, Left-Wing, Right-Wing, Religious, and Single Issue). In order to test the KNN accuracy, I decided to run the model against each of these targets separately, adjust based on the optimal K, and then use a K-Fold Cross Validation as a metric/ test for accuracy. This source for this section was my notes from MSDS650.

1.5.1 KNN Run for Ethnonationalist-Separatist

[17]:
```
features = df.drop(['Event ID#', 'Year', 'Month', 'Day', 'State', 'City',␣
 ↪'Group', 'Dominant Ideology_Ethnonationalist-Separatist', 'Dominant␣
 ↪Ideology_Extreme Left Wing', 'Dominant Ideology_Extreme Right Wing',␣
 ↪'Dominant Ideology_Religious', 'Dominant Ideology_Single Issue', 'Dominant␣
 ↪Ideology_Unknown Ideology'], axis=1)
targets = df['Dominant Ideology_Ethnonationalist-Separatist']
```

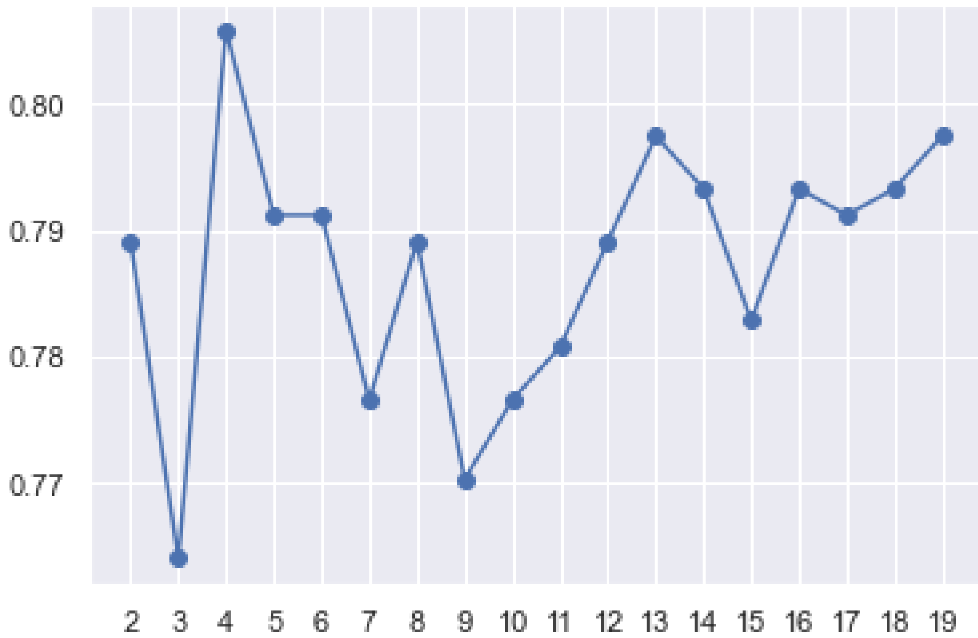
[18]:
```
X_train, X_test, y_train, y_test = train_test_split(features, targets,␣
↪test_size=0.2, random_state=42)
```

[19]:
```
scores = []
for k in range(2, 20):
    print(f'Evaluating {k} clusters')

    model = KNeighborsClassifier(n_neighbors=k, n_jobs=-1)
    model.fit(X_train, y_train)
    scores.append(model.score(X_test, y_test))
```

```
Evaluating 2 clusters
Evaluating 3 clusters
Evaluating 4 clusters
Evaluating 5 clusters
Evaluating 6 clusters
Evaluating 7 clusters
Evaluating 8 clusters
Evaluating 9 clusters
Evaluating 10 clusters
Evaluating 11 clusters
Evaluating 12 clusters
Evaluating 13 clusters
Evaluating 14 clusters
Evaluating 15 clusters
Evaluating 16 clusters
Evaluating 17 clusters
Evaluating 18 clusters
Evaluating 19 clusters
```

[20]:
```
plt.plot(range(2, 20), scores)
plt.scatter(range(2, 20), scores)
plt.xticks(range(2, 20))

print(f'\nMax accuracy = {(max(scores)*100)}%')
```

```

Max accuracy = 80.58455114822547%
```



[21]:
```
model = KNeighborsClassifier(n_neighbors=4, n_jobs=-1)
model.fit(X_train, y_train)
print(model.score(X_train, y_train))
print(model.score(X_test, y_test))
```

```
0.8182767624020888
0.8058455114822547
```

K-Fold Cross Validation for Ethnonationalist-Separatist

[22]:
```
seed = 42
num_folds = 5
scoring = 'accuracy'
```

[23]:
```
ensembles = []
ensembles.append(('KNN', KNeighborsClassifier(n_neighbors=4, n_jobs=-1)))
ensembles.append(('AB', AdaBoostClassifier()))
ensembles.append(('GBM', GradientBoostingClassifier()))
ensembles.append(('RF', RandomForestClassifier(n_estimators=10)))
ensembles.append(('ET', ExtraTreesClassifier(n_estimators=10)))
```

[24]:
```
results = []
names = []
for name, model in ensembles:
    kfold = KFold(n_splits=num_folds, random_state=seed, shuffle=True)
```

```
    cv_results = cross_val_score(model, X_train, y_train, cv=kfold,␣
↪scoring=scoring)
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)
```

```
KNN: 0.778068 (0.020157)
AB: 0.779112 (0.016808)
GBM: 0.785379 (0.018639)
RF: 0.792167 (0.015631)
ET: 0.791123 (0.013617)
```

1.5.2 KNN Run for Extreme Left Wing

[25]:
```
features = df.drop(['Event ID#', 'Year', 'Month', 'Day', 'State', 'City',␣
↪'Group', 'Dominant Ideology_Ethnonationalist-Separatist', 'Dominant␣
↪Ideology_Extreme Left Wing', 'Dominant Ideology_Extreme Right Wing',␣
↪'Dominant Ideology_Religious', 'Dominant Ideology_Single Issue', 'Dominant␣
↪Ideology_Unknown Ideology'], axis=1)
targets = df['Dominant Ideology_Extreme Left Wing']
```

[26]:
```
X_train, X_test, y_train, y_test = train_test_split(features, targets,␣
↪test_size=0.2, random_state=42)
```

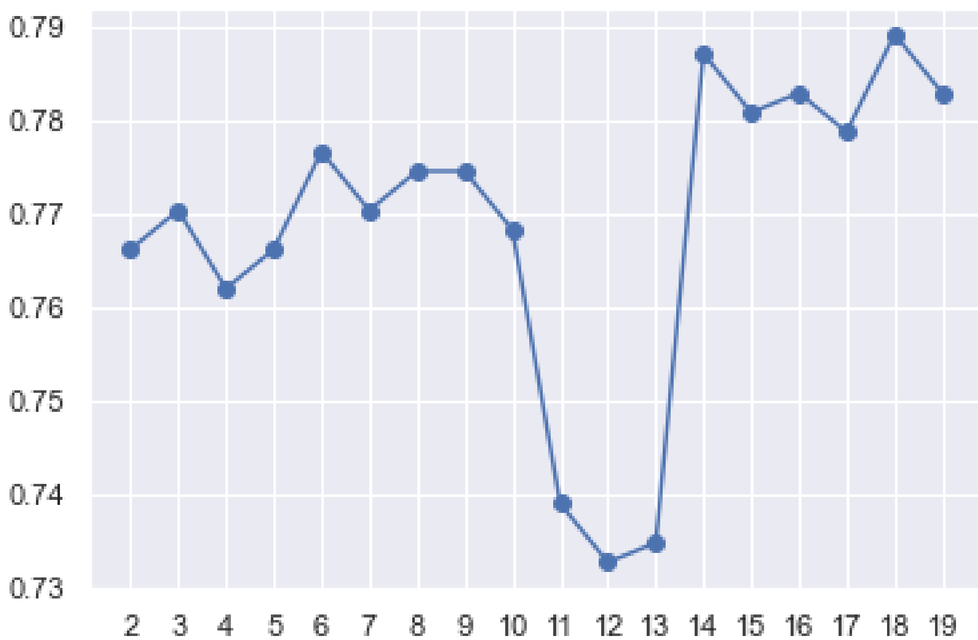
[27]:
```
scores = []
for k in range(2, 20):
    print(f'Evaluating {k} clusters')

    model = KNeighborsClassifier(n_neighbors=k, n_jobs=-1)
    model.fit(X_train, y_train)
    scores.append(model.score(X_test, y_test))
```

```
Evaluating 2 clusters
Evaluating 3 clusters
Evaluating 4 clusters
Evaluating 5 clusters
Evaluating 6 clusters
Evaluating 7 clusters
Evaluating 8 clusters
Evaluating 9 clusters
Evaluating 10 clusters
Evaluating 11 clusters
Evaluating 12 clusters
Evaluating 13 clusters
Evaluating 14 clusters
Evaluating 15 clusters
```

```
Evaluating 16 clusters
Evaluating 17 clusters
Evaluating 18 clusters
Evaluating 19 clusters
```

[28]:
```
plt.plot(range(2, 20), scores)
plt.scatter(range(2, 20), scores)
plt.xticks(range(2, 20))

print(f'\nMax accuracy = {(max(scores)*100)}%')
```

```
Max accuracy = 78.91440501043841%
```



[29]:
```
model = KNeighborsClassifier(n_neighbors=18, n_jobs=-1)
model.fit(X_train, y_train)
print(model.score(X_train, y_train))
print(model.score(X_test, y_test))
```

```
0.8219321148825065
0.7891440501043842
```

K-Fold Cross Validation for Extreme Left Wing

[30]:
```
seed = 42
num_folds = 5
scoring = 'accuracy'
```

[31]:
```
ensembles = []
ensembles.append(('KNN', KNeighborsClassifier(n_neighbors=18, n_jobs=-1)))
ensembles.append(('AB', AdaBoostClassifier()))
ensembles.append(('GBM', GradientBoostingClassifier()))
ensembles.append(('RF', RandomForestClassifier(n_estimators=10)))
ensembles.append(('ET', ExtraTreesClassifier(n_estimators=10)))
```

[32]:
```
results = []
names = []
for name, model in ensembles:
    kfold = KFold(n_splits=num_folds, random_state=seed, shuffle=True)
    cv_results = cross_val_score(model, X_train, y_train, cv=kfold,␣
 ↪scoring=scoring)
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)
```

```
KNN: 0.803133 (0.005376)
AB: 0.796345 (0.011077)
GBM: 0.799478 (0.008800)
RF: 0.807311 (0.006888)
ET: 0.806789 (0.010444)
```

1.5.3 KNN Run for Extreme Right Wing

[33]:
```
features = df.drop(['Event ID#', 'Year', 'Month', 'Day', 'State', 'City',␣
 ↪'Group', 'Dominant Ideology_Ethnonationalist-Separatist', 'Dominant␣
 ↪Ideology_Extreme Left Wing', 'Dominant Ideology_Extreme Right Wing',␣
 ↪'Dominant Ideology_Religious', 'Dominant Ideology_Single Issue', 'Dominant␣
 ↪Ideology_Unknown Ideology'], axis=1)
targets = df['Dominant Ideology_Extreme Right Wing']
```

[34]:
```
X_train, X_test, y_train, y_test = train_test_split(features, targets,␣
 ↪test_size=0.2, random_state=42)
```

[35]:
```
scores = []
for k in range(2, 20):
    print(f'Evaluating {k} clusters')

    model = KNeighborsClassifier(n_neighbors=k, n_jobs=-1)
    model.fit(X_train, y_train)
    scores.append(model.score(X_test, y_test))
```

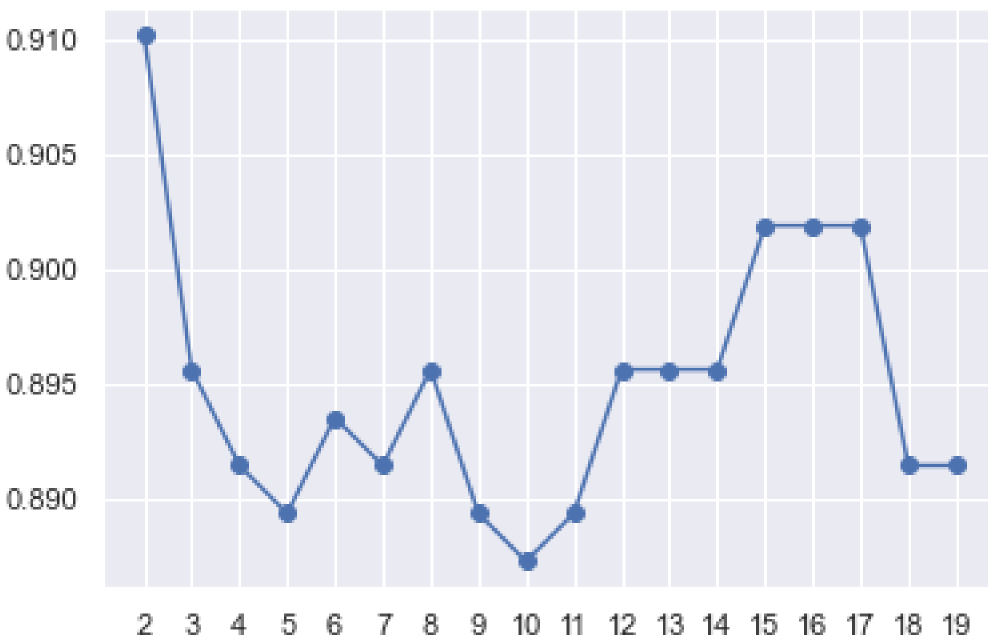
```
Evaluating 2 clusters
Evaluating 3 clusters
Evaluating 4 clusters
```

```
Evaluating 5 clusters
Evaluating 6 clusters
Evaluating 7 clusters
Evaluating 8 clusters
Evaluating 9 clusters
Evaluating 10 clusters
Evaluating 11 clusters
Evaluating 12 clusters
Evaluating 13 clusters
Evaluating 14 clusters
Evaluating 15 clusters
Evaluating 16 clusters
Evaluating 17 clusters
Evaluating 18 clusters
Evaluating 19 clusters
```

[36]:

```
plt.plot(range(2, 20), scores)
plt.scatter(range(2, 20), scores)
plt.xticks(range(2, 20))

print(f'\nMax accuracy = {(max(scores)*100)}%')
```

```
Max accuracy = 91.02296450939458%
```



[37]:
```
model = KNeighborsClassifier(n_neighbors=2, n_jobs=-1)
model.fit(X_train, y_train)
print(model.score(X_train, y_train))
print(model.score(X_test, y_test))
```

```
0.9122715404699739
0.9102296450939458
```

K-Fold Cross Validation for Extreme Right Wing

[38]:
```
seed = 42
num_folds = 5
scoring = 'accuracy'
```

[39]:
```
ensembles = []
ensembles.append(('KNN', KNeighborsClassifier(n_neighbors=2, n_jobs=-1)))
ensembles.append(('AB', AdaBoostClassifier()))
ensembles.append(('GBM', GradientBoostingClassifier()))
ensembles.append(('RF', RandomForestClassifier(n_estimators=10)))
ensembles.append(('ET', ExtraTreesClassifier(n_estimators=10)))
```

[40]:
```
results = []
names = []
for name, model in ensembles:
    kfold = KFold(n_splits=num_folds, random_state=seed, shuffle=True)
    cv_results = cross_val_score(model, X_train, y_train, cv=kfold,␣
 ↪scoring=scoring)
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)
```

```
KNN: 0.875718 (0.019857)
AB: 0.879373 (0.015080)
GBM: 0.886684 (0.013272)
RF: 0.885117 (0.013816)
ET: 0.887206 (0.010099)
```

1.5.4 KNN Run for Religious

[41]:
```
features = df.drop(['Event ID#', 'Year', 'Month', 'Day', 'State', 'City',␣
 ↪'Group', 'Dominant Ideology_Ethnonationalist-Separatist', 'Dominant␣
 ↪Ideology_Extreme Left Wing', 'Dominant Ideology_Extreme Right Wing',␣
 ↪'Dominant Ideology_Religious', 'Dominant Ideology_Single Issue', 'Dominant␣
 ↪Ideology_Unknown Ideology'], axis=1)
targets = df['Dominant Ideology_Religious']
```

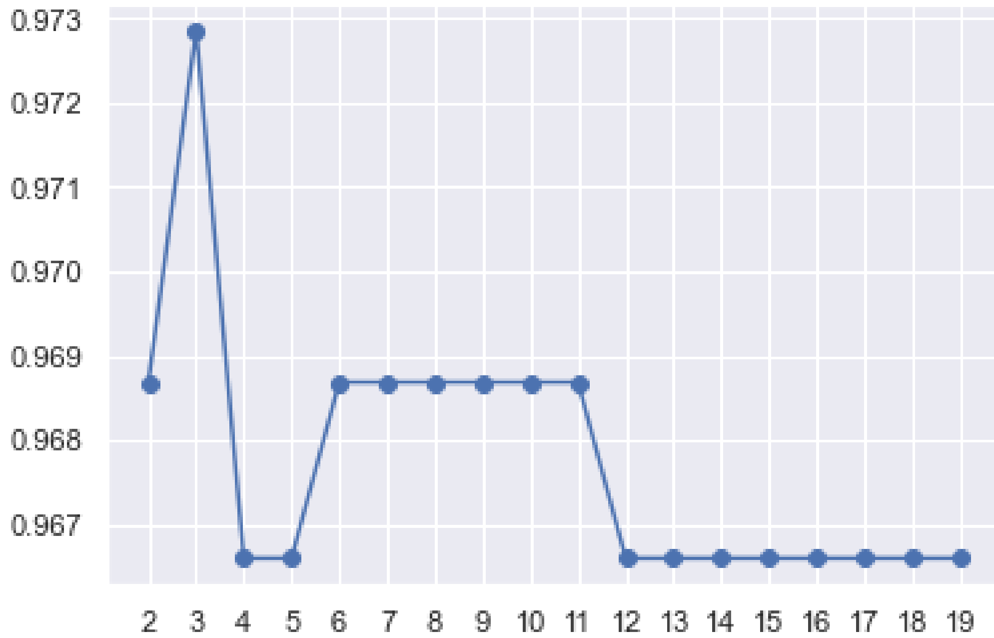
[42]:
```python
X_train, X_test, y_train, y_test = train_test_split(features, targets,␣
↪test_size=0.2, random_state=42)
```

[43]:
```python
scores = []
for k in range(2, 20):
    print(f'Evaluating {k} clusters')

    model = KNeighborsClassifier(n_neighbors=k, n_jobs=-1)
    model.fit(X_train, y_train)
    scores.append(model.score(X_test, y_test))
```

```
Evaluating 2 clusters
Evaluating 3 clusters
Evaluating 4 clusters
Evaluating 5 clusters
Evaluating 6 clusters
Evaluating 7 clusters
Evaluating 8 clusters
Evaluating 9 clusters
Evaluating 10 clusters
Evaluating 11 clusters
Evaluating 12 clusters
Evaluating 13 clusters
Evaluating 14 clusters
Evaluating 15 clusters
Evaluating 16 clusters
Evaluating 17 clusters
Evaluating 18 clusters
Evaluating 19 clusters
```

[44]:
```python
plt.plot(range(2, 20), scores)
plt.scatter(range(2, 20), scores)
plt.xticks(range(2, 20))

print(f'\nMax accuracy = {(max(scores)*100)}%')
```

```
Max accuracy = 97.28601252609603%
```



[45]:
```
model = KNeighborsClassifier(n_neighbors=3, n_jobs=-1)
model.fit(X_train, y_train)
print(model.score(X_train, y_train))
print(model.score(X_test, y_test))
```

```
0.970757180156658
0.9728601252609603
```

K-Fold Cross Validation for Religious

[46]:
```
seed = 42
num_folds = 5
scoring = 'accuracy'
```

[47]:
```
ensembles = []
ensembles.append(('KNN', KNeighborsClassifier(n_neighbors=3, n_jobs=-1)))
ensembles.append(('AB', AdaBoostClassifier()))
ensembles.append(('GBM', GradientBoostingClassifier()))
ensembles.append(('RF', RandomForestClassifier(n_estimators=10)))
ensembles.append(('ET', ExtraTreesClassifier(n_estimators=10)))
```

[48]:
```
results = []
names = []
for name, model in ensembles:
    kfold = KFold(n_splits=num_folds, random_state=seed, shuffle=True)
```

```
    cv_results = cross_val_score(model, X_train, y_train, cv=kfold,␣
↪scoring=scoring)
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)
```

```
KNN: 0.959269 (0.010778)
AB: 0.955091 (0.009543)
GBM: 0.963969 (0.009105)
RF: 0.960836 (0.008893)
ET: 0.962924 (0.007083)
```

1.5.5 KNN Run for Single Issue

[49]:
```
features = df.drop(['Event ID#', 'Year', 'Month', 'Day', 'State', 'City',␣
↪'Group', 'Dominant Ideology_Ethnonationalist-Separatist', 'Dominant␣
↪Ideology_Extreme Left Wing', 'Dominant Ideology_Extreme Right Wing',␣
↪'Dominant Ideology_Religious', 'Dominant Ideology_Single Issue', 'Dominant␣
↪Ideology_Unknown Ideology'], axis=1)
targets = df['Dominant Ideology_Single Issue']
```

[50]:
```
X_train, X_test, y_train, y_test = train_test_split(features, targets,␣
↪test_size=0.2, random_state=42)
```

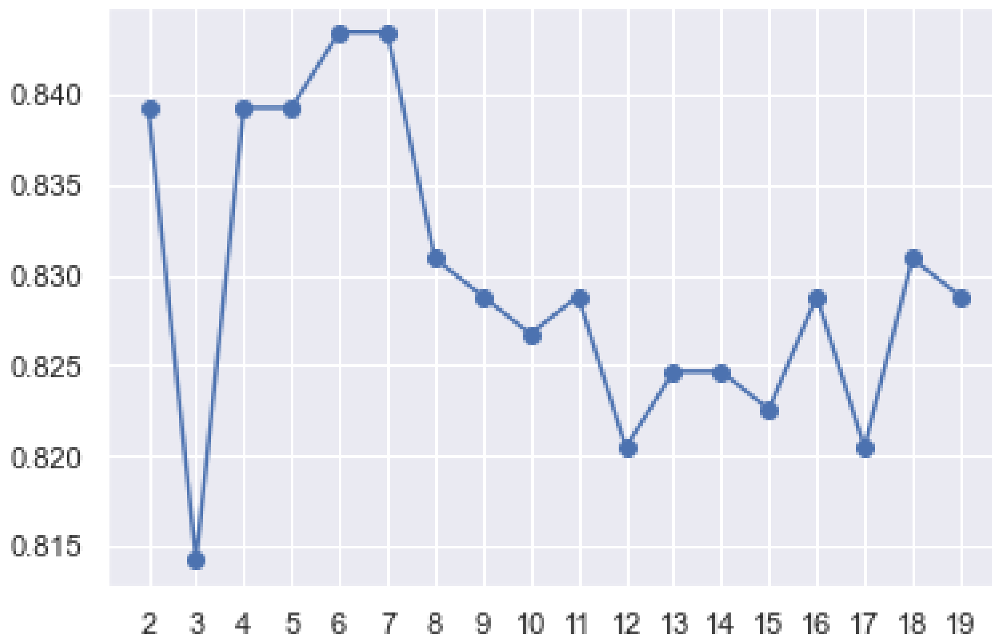
[51]:
```
scores = []
for k in range(2, 20):
    print(f'Evaluating {k} clusters')

    model = KNeighborsClassifier(n_neighbors=k, n_jobs=-1)
    model.fit(X_train, y_train)
    scores.append(model.score(X_test, y_test))
```

```
Evaluating 2 clusters
Evaluating 3 clusters
Evaluating 4 clusters
Evaluating 5 clusters
Evaluating 6 clusters
Evaluating 7 clusters
Evaluating 8 clusters
Evaluating 9 clusters
Evaluating 10 clusters
Evaluating 11 clusters
Evaluating 12 clusters
Evaluating 13 clusters
Evaluating 14 clusters
Evaluating 15 clusters
```

```
Evaluating 16 clusters
Evaluating 17 clusters
Evaluating 18 clusters
Evaluating 19 clusters
```

[52]:
```
plt.plot(range(2, 20), scores)
plt.scatter(range(2, 20), scores)
plt.xticks(range(2, 20))

print(f'\nMax accuracy = {(max(scores)*100)}%')
```

```
Max accuracy = 84.34237995824635%
```



[53]:
```
model = KNeighborsClassifier(n_neighbors=7, n_jobs=-1)
model.fit(X_train, y_train)
print(model.score(X_train, y_train))
print(model.score(X_test, y_test))
```

```
0.8605744125326371
0.8434237995824635
```

K-Fold Cross Validation for Single Issue

[54]:
```
seed = 42
num_folds = 5
scoring = 'accuracy'
```

```
[55]: ensembles = []
ensembles.append(('KNN', KNeighborsClassifier(n_neighbors=7, n_jobs=-1)))
ensembles.append(('AB', AdaBoostClassifier()))
ensembles.append(('GBM', GradientBoostingClassifier()))
ensembles.append(('RF', RandomForestClassifier(n_estimators=10)))
ensembles.append(('ET', ExtraTreesClassifier(n_estimators=10)))
```

```
[56]: results = []
names = []
for name, model in ensembles:
    kfold = KFold(n_splits=num_folds, random_state=seed, shuffle=True)
    cv_results = cross_val_score(model, X_train, y_train, cv=kfold,␣
 ↪scoring=scoring)
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)
```

```
KNN: 0.829765 (0.023493)
AB: 0.831332 (0.006310)
GBM: 0.836554 (0.008355)
RF: 0.829765 (0.014621)
ET: 0.841253 (0.014051)
```

1.6 Final Analysis and Further Research

The accuracy for the KNN runs were as follows:

- Ethnonationalist-Separatist: 80.58%
- Extreme Left Wing: 78.91%
- Extreme Right Wing: 91.02%
- Religious: 97.28%
- Single Issue: 84.34%

  The high number for accurately identifying Religously motivated attacks gives me pause and merits further investigation. It could be that accurate, but, based on my limited experience in MSDS650, I doubt it. There may be a feature that is triggering this level of accuracy and allowing the machine to "cheat." I experienced this phenonmena during initial tests when I had some extra sub-ideology features that were actually giving the machine the answer to the target.

  The K-Fold cross validations showed a slight improvement using Extra Trees or Random Forests. This may merit some experimentation but the gains were minimal.

For further research, more detailed data on the incidents would increase the confidence in the model outcomes. For example, detailed data on attackers/ perpetrators might be another variable that could assist in classification.

```
[ ]:
```