Predict Arrests Made After Terry Stop

Author: Carlos Garza

Overview

This notebook details the building, tuning and deployment of a categorical model that predicts whether an arrest was made during traffic stops by the Seattle Police. Tools utilized include but are not limited to K-Nearest Neighbors models, decision trees, random forests, and XGBoost.

Because the data includes race and gender of both the officers and suspects, this will be a purely objective model based on public data and will not include further social commentary. Social issues regarding race and gender are outside of the scope of this project.

Business Problem

In the late 1960's, the supreme court ruled in Terry vs. Ohio that "stop and frisk" police tactics are not a violation of constitutional rights. Because of this, police can detain a person on the grounds of "reasonable suspicion," even in the absence of clearer evidence.

This ruling lead to the coining of the term Terry Stop, which is when an officer of the law briefly detains a driver based on the reasonable suspician that the driver is involved in criminal activity.

Based on data from Terry Stops from the Seattle Police Department, I built a classifying model that predicts if a stop will result in an arrest.

Data

The data utilized in this project comes from data.seattle.gov and consists of 46.3k Terry Stops, as reported by the conducting officer. A copy of the csv file can be found in the data folder of this repository or at data.gov.

Exploratory Data Analysis

Exploring the data to develop a preprocessing and modeling strategy

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:
```
df = pd.read_csv('data/Terry_Stops.csv')
df.head()
```

Out[2]:

	Subject Age Group	Subject ID	GO / SC Num	Terry Stop ID	Stop Resolution	Weapon Type	Officer ID	Officer YOB	Officer Gender	O
0	-	-1	20140000120677	92317	Arrest	None	7500	1984	M	Bla A Ame
1	-	-1	20150000001463	28806	Field Contact	None	5670	1965	M	
2	-	-1	20150000001516	29599	Field Contact	None	4844	1961	M	
3	-	-1	20150000001670	32260	Field Contact	None	7539	1963	M	
4	-	-1	20150000001739	33155	Field Contact	None	6973	1977	M	

5 rows × 23 columns

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 46248 entries, 0 to 46247
Data columns (total 23 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   Subject Age Group         46248 non-null  object
 1   Subject ID                46248 non-null  int64 
 2   GO / SC Num               46248 non-null  int64 
 3   Terry Stop ID             46248 non-null  int64 
 4   Stop Resolution           46248 non-null  object
 5   Weapon Type               46248 non-null  object
 6   Officer ID                46248 non-null  object
 7   Officer YOB               46248 non-null  int64 
 8   Officer Gender            46248 non-null  object
 9   Officer Race              46248 non-null  object
 10  Subject Perceived Race    46248 non-null  object
 11  Subject Perceived Gender  46248 non-null  object
 12  Reported Date             46248 non-null  object
 13  Reported Time             46248 non-null  object
 14  Initial Call Type         46248 non-null  object
 15  Final Call Type           46248 non-null  object
 16  Call Type                 46248 non-null  object
 17  Officer Squad             45643 non-null  object
 18  Arrest Flag               46248 non-null  object
 19  Frisk Flag                46248 non-null  object
 20  Precinct                  46248 non-null  object
 21  Sector                    46248 non-null  object
 22  Beat                      46248 non-null  object
dtypes: int64(4), object(19)
memory usage: 8.1+ MB
```

In [4]:
```
for column in df.columns:
    print(column, '\n')
    print(df[column].value_counts())
    print('__________')
```

```
Subject Age Group 

26 - 35          15381
```

```
36 - 45          9762
18 - 25          9322
46 - 55          5980
56 and Above     2345
1 - 17           1961
-                1497
Name: Subject Age Group, dtype: int64
_________
Subject ID

-1              34742
 7726859935        19
 7753260438        18
 7727117712        13
 12795904212       11
                ...
 7728382188         1
 7728607474         1
 7725672697         1
 7725797630         1
 16219707395        1
Name: Subject ID, Length: 8632, dtype: int64
_________
GO / SC Num

20160000378750    16
20150000190790    16
20180000134604    14
20190000441736    13
20170000132836    13
                  ..
20190000410715     1
20160000174160     1
20170000156685     1
20200000272466     1
20180000071981     1
Name: GO / SC Num, Length: 36213, dtype: int64
_________
Terry Stop ID

19268585233    3
19324329995    3
13080077761    3
15045077325    3
12119304761    2
              ..
76432          1
250511         1
383630         1
49805          1
131072         1
Name: Terry Stop ID, Length: 46213, dtype: int64
_________
Stop Resolution

Field Contact               18520
Offense Report              15437
Arrest                      11386
Referred for Prosecution      728
Citation / Infraction         177
Name: Stop Resolution, dtype: int64
_________
Weapon Type

None                                    32565
```

```
-                                         10972
Lethal Cutting Instrument                  1482
Knife/Cutting/Stabbing Instrument           574
Handgun                                     284
Firearm Other                               100
Blunt Object/Striking Implement              76
Club, Blackjack, Brass Knuckles              49
Firearm                                      36
Mace/Pepper Spray                            27
Other Firearm                                22
Firearm (unk type)                           15
Taser/Stun Gun                                9
Club                                          9
Rifle                                         7
None/Not Applicable                           7
Fire/Incendiary Device                        6
Shotgun                                       3
Automatic Handgun                             2
Personal Weapons (hands, feet, etc.)          1
Brass Knuckles                                1
Blackjack                                     1
Name: Weapon Type, dtype: int64
_________
Officer ID

7456      415
7634      341
7773      321
7765      315
7758      308
         ...
8786        1
5445        1
5137        1
7558        1
5411        1
Name: Officer ID, Length: 1199, dtype: int64
_________
Officer YOB

1986    3236
1987    2958
1984    2721
1991    2658
1985    2477
1992    2377
1990    2204
1988    2056
1989    1953
1982    1835
1983    1700
1979    1488
1993    1417
1981    1398
1971    1217
1978    1148
1995    1099
1976     999
1977     991
1973     914
1994     880
1980     798
1967     708
1968     625
1996     595
```

```
1970     589
1974     556
1969     535
1975     525
1962     455
1972     420
1965     416
1964     415
1997     370
1963     258
1966     223
1958     218
1961     212
1959     174
1960     161
1900      71
1954      44
1957      43
1953      32
1955      21
1956      17
1998      13
1948      11
1952       9
1949       5
1946       2
1951       1
Name: Officer YOB, dtype: int64
_________
Officer Gender

M    40933
F     5286
N       29
Name: Officer Gender, dtype: int64
_________
Officer Race

White                            35036
Hispanic or Latino                2646
Two or More Races                 2590
Asian                             1967
Black or African American         1828
Not Specified                     1346
Nat Hawaiian/Oth Pac Islander      447
American Indian/Alaska Native      317
Unknown                             71
Name: Officer Race, dtype: int64
_________
Subject Perceived Race

White                                        22572
Black or African American                    13760
Unknown                                       2519
-                                             1883
Hispanic                                      1684
Asian                                         1487
American Indian or Alaska Native              1334
Multi-Racial                                   809
Other                                          152
Native Hawaiian or Other Pacific Islander       48
Name: Subject Perceived Race, dtype: int64
_________
Subject Perceived Gender
```

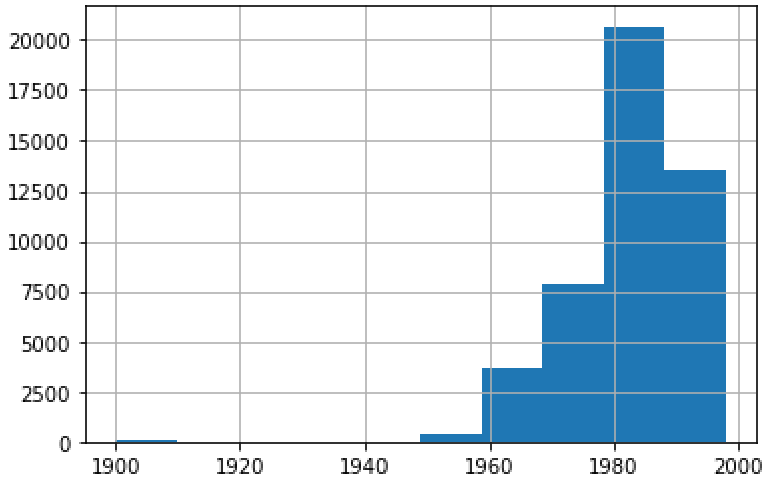
```
Male                                                        36203
Female                                                       9419
Unable to Determine                                           326
-                                                             274
Unknown                                                        22
Gender Diverse (gender non-conforming and/or transgender)       4
Name: Subject Perceived Gender, dtype: int64
_________
Reported Date

2015-10-01T00:00:00    101
2015-09-29T00:00:00     66
2015-05-28T00:00:00     57
2015-07-18T00:00:00     55
2019-04-26T00:00:00     54
                      ...
2015-05-10T00:00:00      1
2015-03-15T00:00:00      1
2015-03-28T00:00:00      1
2015-03-24T00:00:00      1
2015-03-31T00:00:00      1
Name: Reported Date, Length: 2159, dtype: int64
_________
Reported Time

02:56:00    51
17:00:00    51
19:18:00    51
03:09:00    50
18:51:00    50
            ..
16:32:29     1
13:10:10     1
23:42:50     1
01:34:36     1
09:53:36     1
Name: Reported Time, Length: 12158, dtype: int64
_________
Initial Call Type

-                                                 13110
SUSPICIOUS STOP - OFFICER INITIATED ONVIEW         3091
SUSPICIOUS PERSON, VEHICLE OR INCIDENT             2917
DISTURBANCE, MISCELLANEOUS/OTHER                   2375
ASLT - IP/JO - WITH OR W/O WPNS (NO SHOOTINGS)     1950
                                                  ...
WARRANT PICKUP - FROM OTHER AGENCY                    1
DEMONSTRATIONS                                        1
MISSING - ADULT                                       1
MISSING - (ALZHEIMER, ENDANGERED, ELDERLY)            1
INJURED -  PERSON/INDUSTRIAL ACCIDENT                 1
Name: Initial Call Type, Length: 167, dtype: int64
_________
Final Call Type

-                                                     13110
--SUSPICIOUS CIRCUM. - SUSPICIOUS PERSON               3660
--PROWLER - TRESPASS                                   3264
--DISTURBANCE - OTHER                                  2655
--ASSAULTS, OTHER                                      2239
                                                      ...
--PREMISE CHECKS - REQUEST TO WATCH                       1
BIAS -IP/JO - RACIAL, POLITICAL, SEXUAL MOTIVATION        1
--HARBOR - ASSIST BOATER (NON EMERG)                      1
PROWLER                                                   1
```

```
--COMMERCIAL SEXUAL EXPLOITATION OF MINORS (CSEC)         1
Name: Final Call Type, Length: 205, dtype: int64
_________
Call Type

911                               20700
-                                 13110
ONVIEW                             8891
TELEPHONE OTHER, NOT 911           3224
ALARM CALL (NOT POLICE ALARM)       315
TEXT MESSAGE                          7
SCHEDULED EVENT (RECURRING)           1
Name: Call Type, dtype: int64
_________
Officer Squad

TRAINING - FIELD TRAINING SQUAD                     4951
WEST PCT 1ST W - DAVID/MARY                         1525
WEST PCT 2ND W - D/M RELIEF                         1003
SOUTHWEST PCT 2ND W - FRANK                          942
NORTH PCT 2ND WATCH - NORTH BEATS                    885
                                                    ...
ZOLD CRIME ANALYSIS UNIT - ANALYSTS                    1
COMMUNITY OUTREACH - SPECIAL PROJECTS DETAIL           1
SOUTHWEST PCT OPS - BURG/THEFT                         1
TRAINING - ADVANCED - SQUAD C                          1
DV SQUAD D - ORDER SERVICE                             1
Name: Officer Squad, Length: 170, dtype: int64
_________
Arrest Flag

N    43070
Y     3178
Name: Arrest Flag, dtype: int64
_________
Frisk Flag

N    35438
Y    10332
-      478
Name: Frisk Flag, dtype: int64
_________
Precinct

West         11096
North        10172
-             9806
East          6105
South         5542
Southwest     2320
SouthWest      956
Unknown        200
OOJ             32
FK ERROR        19
Name: Precinct, dtype: int64
_________
Sector

-         10010
E          2337
M          2270
N          2191
K          1762
B          1658
L          1639
```

```
K           1618
D           1512
R           1455
F           1378
S           1348
U           1302
M           1161
O           1161
J           1119
G           1087
D           1069
C           1037
Q            967
W            941
E            879
Q            733
N            657
F            574
O            566
R            563
B            462
S            450
U            420
G            417
W            382
J            368
L            355
C            347
99            53
Name: Sector, dtype: int64
_________
Beat

-          9951
N3         1175
E2         1092
K3          905
M2          852
           ...
N1           71
99           53
99           31
OOJ          22
S             2
Name: Beat, Length: 107, dtype: int64
_________
```

In [5]:
```
df['Officer YOB'].hist()
```

Out[5]: <AxesSubplot:>



Data Preprocessing

To begin I dropped the columns that identify specific suspects or officers, as well as locational variables to make a more generalized dataset. Additionally, call types were dropped as the categorical data is expansive.

In [6]:
```
to_drop = ['Subject ID', 'GO / SC Num', 'Terry Stop ID', 'Officer ID',
           'Initial Call Type', 'Final Call Type', 'Call Type', 'Officer Squad',
           'Precinct', 'Sector', 'Beat']
```

In [7]:
```
df.drop(to_drop, axis=1, inplace=True)
df.head()
```

Out[7]:

	Subject Age Group	Stop Resolution	Weapon Type	Officer YOB	Officer Gender	Officer Race	Subject Perceived Race	Subject Perceived Gender	Reported Date	I
0	-	Arrest	None	1984	M	Black or African American	Asian	Male	2015-10-16T00:00:00	
1	-	Field Contact	None	1965	M	White	-	-	2015-03-19T00:00:00	
2	-	Field Contact	None	1961	M	White	White	Male	2015-03-21T00:00:00	
3	-	Field Contact	None	1963	M	White	-	-	2015-04-01T00:00:00	
4	-	Field Contact	None	1977	M	White	Black or African American	Male	2015-04-03T00:00:00	

In the following lines I use the 'Stop Resolution' and 'Arrest Flag' columns to create a binary 'Arrested' column.

In [8]:
```
def define_arrested(resolution, flag):
    if resolution == 'Arrest' or flag == 'Y':
        return 1
    else:
        return 0
```

In [9]:
```
df['Arrested'] = df.apply(lambda x: define_arrested(x['Stop Resolution'], x['Arr
df.drop(['Stop Resolution', 'Arrest Flag'], axis=1, inplace=True)
```

For 'Frisk Flag and 'Weapon Type' columns, I will assume that missing data signifies no frisk/weapon.

In [10]:
```
df['Frisk Flag'] = df['Frisk Flag'].map(lambda x: 1 if x == 'Y' else 0)
```

In [11]:
```
df['Weapon Type'] = df['Weapon Type'].map(lambda x: 'None' if x == '-' else x)
```

For date and time, the data will be converted to numerical 'month' and 'hour' columns respecively.

In [12]:
```
df['Reported Month'] = df['Reported Date'].map(lambda x: int(x[5:7]))
df.drop('Reported Date', axis=1, inplace=True)
```

In [13]:
```
df['Reported Hour'] = df['Reported Time'].map(lambda x: int(x[:2]))
df.drop('Reported Time', axis = 1, inplace=True)
```

Outliers and generic place-holder answers are dropped.

In [14]:
```
df.drop(df[df['Officer YOB']<1940].index, inplace=True)
df.drop(df[df['Subject Perceived Gender']=='-'].index, inplace=True)
df.drop(df[df['Subject Perceived Race']=='-'].index, inplace=True)
df.drop(df[df['Subject Age Group']=='-'].index, inplace=True)
```

To be sure no collinearity problems will arise, the correlation between dependent variables needs to be checked.

Alpha = 0.05

In [15]:
```
df.drop('Arrested', axis=1).corr()
```

Out[15]:

	Officer YOB	Frisk Flag	Reported Month	Reported Hour
Officer YOB	1.000000	0.027183	-0.019369	-0.045488
Frisk Flag	0.027183	1.000000	0.008921	0.018031
Reported Month	-0.019369	0.008921	1.000000	-0.000423
Reported Hour	-0.045488	0.018031	-0.000423	1.000000

Lastly, data is separated into training and testing sets and the categorical variables will be one hot encoded.

In [16]:
```
from sklearn.model_selection import train_test_split

y = df.Arrested
X = df.drop('Arrested', axis=1)
X = pd.get_dummies(X, drop_first=True)
```

In [17]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=40)
```

Baseline Models

Models that will be auditioned are K Nearest Neighbors, decision trees, random forest, and XGBoost

In [18]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score
```

K Nearest Neighbor

In [19]:
```
scaler = StandardScaler()

# Transform the training and test sets
scaled_X_train = scaler.fit_transform(X_train)
scaled_X_test = scaler.transform(X_test)
```

In [20]:
```
knn = KNeighborsClassifier()
knn.fit(scaled_X_train, y_train)
y_train_pred = knn.predict(scaled_X_train)
y_test_pred = knn.predict(scaled_X_test)
print('accuracy score: ', accuracy_score(y_test, y_test_pred))
```

```
accuracy score:  0.7068949560388709
```

Decision Tree

In [21]:
```
dsc = DecisionTreeClassifier(max_depth=3)
dsc.fit(X_train, y_train)
y_pred = dsc.predict(X_test)
print('accuracy score: ', accuracy_score(y_test, y_pred))
```

```
accuracy score:  0.7434521055067098
```

Random Forest

In [22]:
```
rfc = RandomForestClassifier(max_depth=3)
rfc.fit(X_train, y_train)
y_pred = rfc.predict(X_test)
print('accuracy score: ', accuracy_score(y_test, y_pred))
```

```
accuracy score:  0.7445627024525683
```

XGBoost

In [23]:
```
xgb = XGBClassifier()
xgb.fit(X_train, y_train)
y_pred = xgb.predict(X_test)
print('accuracy score: ', accuracy_score(y_test, y_pred))
```

```
accuracy score:  0.7409532623785284
```

Because our decision tree and random forest performed best, they will be focused on moving forward.

Model Tuning/Reitteration

GridSearchCV will be used to tune the hyperparameters of the algorithms to maximize their performance.

Decision Tree

In [24]:
```
from sklearn.model_selection import GridSearchCV
```

In [25]:
```
dtc = DecisionTreeClassifier()

dt_param_grid = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [None, 2, 3, 4, 5, 6],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 3, 4, 5, 6]
}

gs_tree = GridSearchCV(dtc, dt_param_grid, cv=3)
gs_tree.fit(X_train, y_train)
gs_tree.best_params_
```

Out[25]:
```
{'criterion': 'gini',
 'max_depth': 3,
 'min_samples_leaf': 1,
 'min_samples_split': 2}
```

In [26]:
```
gs_tree.score(X_test, y_test)
```

Out[26]:
```
0.7434521055067098
```

In [27]:
```
rfc = RandomForestClassifier()

rf_param_grid = {
    'n_estimators': [10, 30, 100],
    'criterion': ['gini', 'entropy'],
    'max_depth': [None, 2, 4, 6, 10],
    'min_samples_split': [5, 10],
    'min_samples_leaf': [3, 6]
}

gs_forest = GridSearchCV(rfc, rf_param_grid, cv=3)
gs_forest.fit(X_train, y_train)
gs_forest.best_params_
```

Out[27]:
```
{'criterion': 'entropy',
 'max_depth': None,
 'min_samples_leaf': 3,
 'min_samples_split': 5,
 'n_estimators': 100}
```

In [28]:
```
gs_forest.score(X_test, y_test)
```

Out[28]:
```
0.7434521055067098
```

Accuracy is slightly improved after optimization, but only by a negligible amount. Models must be further evaluated to find the model with the best performance.

Model Evaluation

The two contesting models will be further evaluated to narrow down to the best performing model for deployment.

In [29]:
```
from sklearn.metrics import plot_confusion_matrix, recall_score, precision_score

def metric_scores(actual, predicted):
    print('Accuracy Score: ', accuracy_score(actual, predicted))
    print('Recall Score: ', recall_score(actual, predicted))
    print('Precision Score: ', precision_score(actual, predicted))
    print('F1 Score: ', f1_score(actual, predicted))
```

Decision Tree

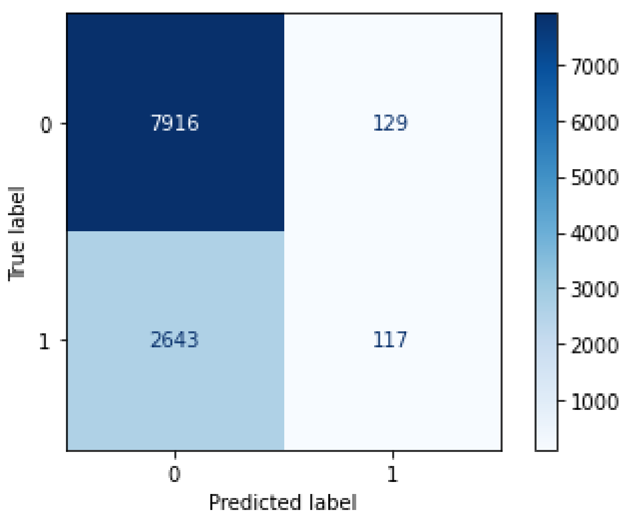
In [30]:
```
dtree = DecisionTreeClassifier(criterion='gini', max_depth=3, min_samples_leaf=1
dtree.fit(X_train, y_train)
tree_pred = dtree.predict(X_test)
```

In [31]:
```
metric_scores(y_test, tree_pred)
```

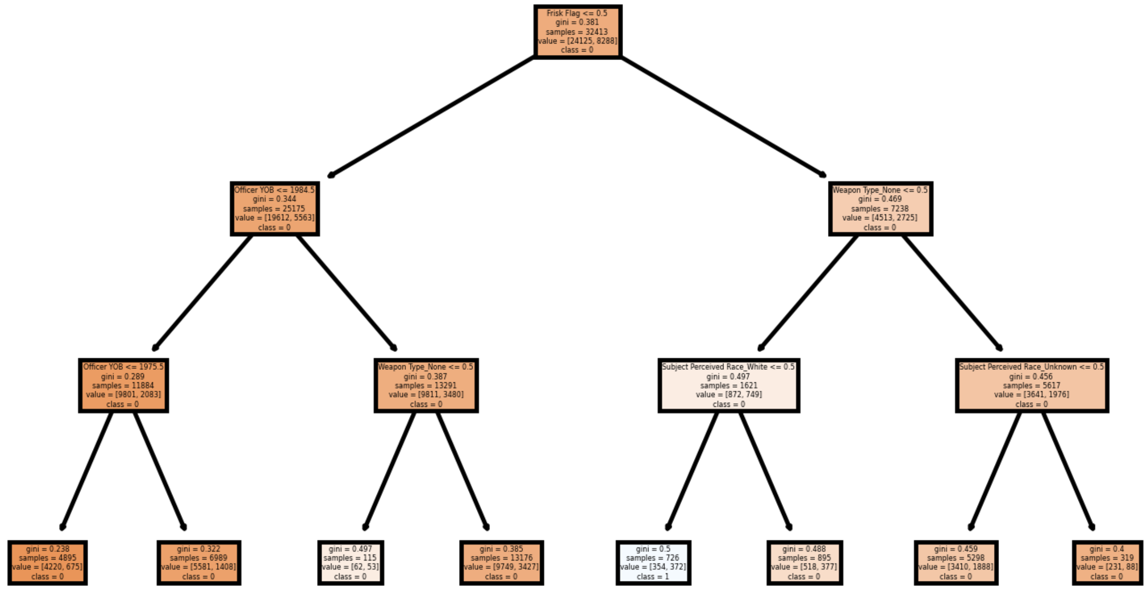
```
Accuracy Score:  0.7434521055067098
Recall Score:  0.042391304347826085
Precision Score:  0.47560975609756095
F1 Score:  0.0778443113772455
```

In [32]:
```
plot_confusion_matrix(dtree, X_test, y_test, cmap=plt.cm.Blues)
```

Out[32]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7ffa0046e040
>
```



In [33]:
```
from sklearn import tree
fig, axes = plt.subplots(nrows = 1,ncols = 1, figsize = (5,3), dpi=350)
tree.plot_tree(dtree,
               feature_names = X_train.columns,
               class_names=np.unique(y_train).astype('str'),
               filled = True)
plt.show()
```



The tree is not overfitted, but does have a large amount of false negative predictions.

Random Forest

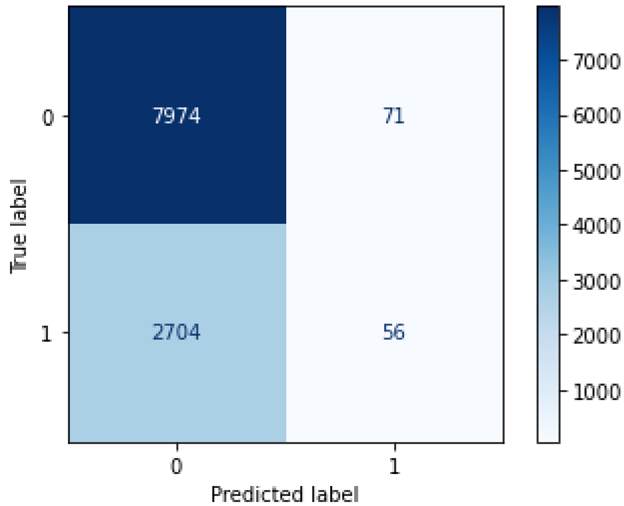
In [34]:
```
rforest = RandomForestClassifier(criterion='gini', max_depth=None, min_samples_l
                                 min_samples_split=10, n_estimators=100)
rforest.fit(X_train, y_train)
forest_pred = rforest.predict(X_test)
```

In [35]:
```
metric_scores(y_test, forest_pred)
```

```
Accuracy Score:  0.7431744562702453
Recall Score:  0.020289855072463767
Precision Score:  0.4409448818897638
F1 Score:  0.038794596466920676
```

In [36]:
```
plot_confusion_matrix(rforest, X_test, y_test, cmap=plt.cm.Blues)
```

Out[36]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7ffa2a80f2e0
>
```

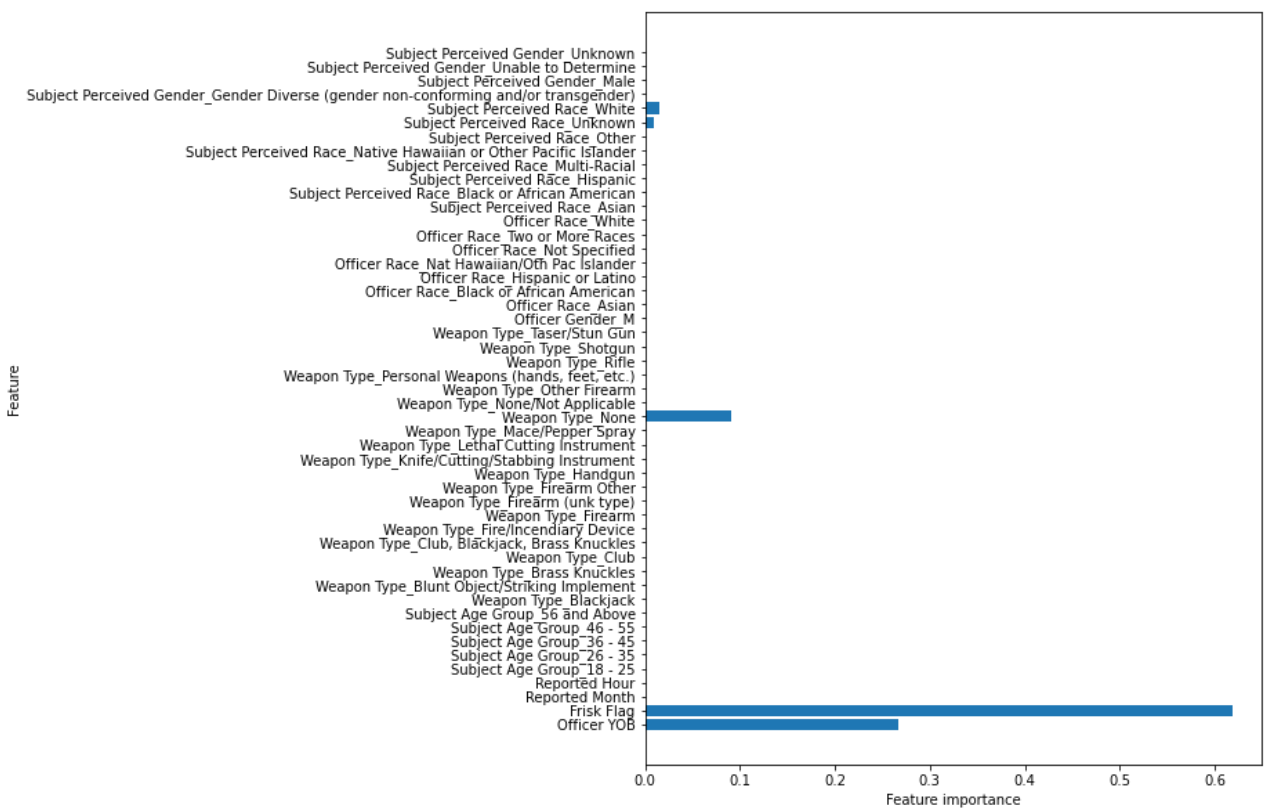


The random forest model performed similarly to the decision tree. To save processing time and power, the final model is the decision tree.

Feature Importance

In [37]:
```
plt.figure(figsize=(8,10))
plt.barh(range(len(X_train.columns)), dtree.feature_importances_, align='center'
plt.yticks(np.arange(len(X_train.columns)), X_train.columns.values)
plt.xlabel('Feature importance')
plt.ylabel('Feature')
```

Out[37]: `Text(0, 0.5, 'Feature')`



Conclusions

The most important feature to a random stop influencing if a subject is arrested is whether or not the subject was frisked. Following that is the age of the officer and whether or not the subject had any weapons. the type of weapon did not matter.

The decision tree classifier trained with this data had an accuracy rating of about 75%

Future Work

In the future it would be beneficial to explore ways to decrease the rate of false positives in our model.

Also, incorporating data from a wider geographic area could make for a more comprehensive model.