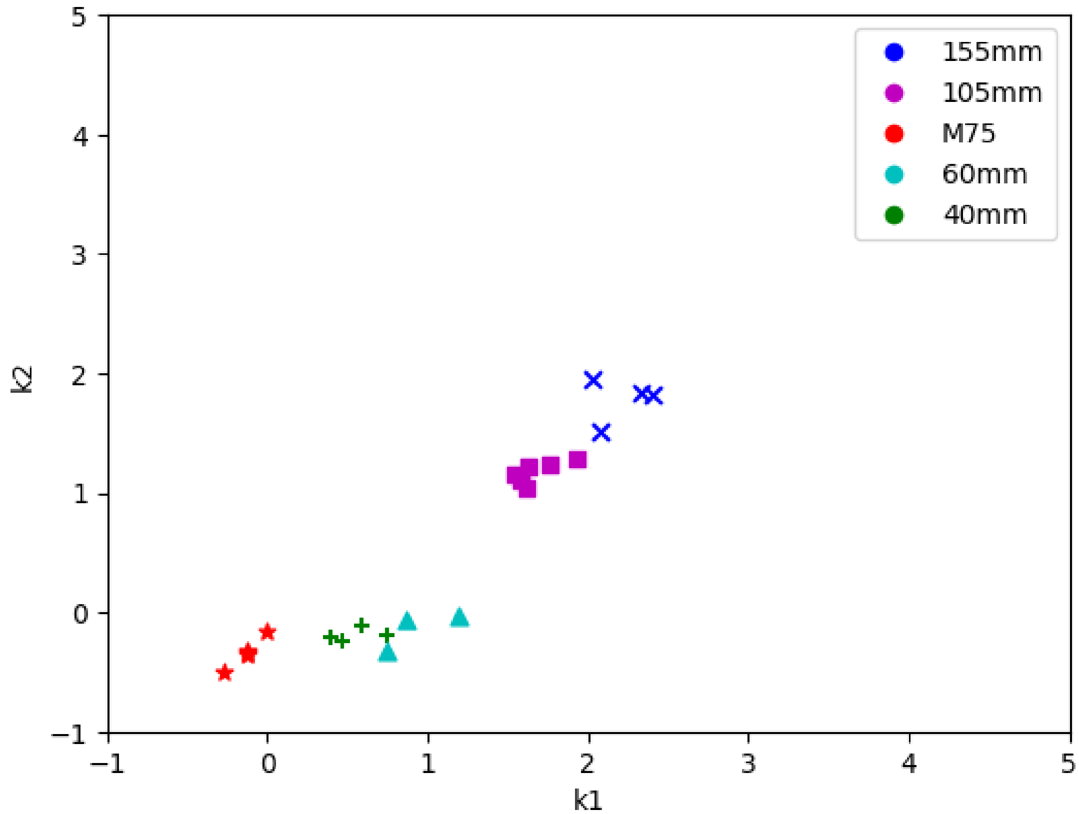
```python
# Import required packages:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import ConfusionMatrixDisplay
from plotGNB import plotDecisionBoundaries

def groupScatter(groups,xlim,ylim):
   ax = plt.gca() # a "handle" to the axes into which you want to add
the scatter plotted points
   # Below, each iteration of the for loop picks out one of the groups
of munitions
   # and plots it using the colours and markers that we define in a
code cell further below.
   for key, group in groups:
      group.plot(ax=ax, kind='scatter', x='k1', y='k2', s=40,
label=key, color=colors[key], marker=markers[key] )
   plt.xlabel('k1')
   plt.ylabel('k2')
   plt.xlim(xlim)
   plt.ylim(ylim)

# This is a good place to define some constants for use later,
# like the "deg" variable you used in the previous assignment to
define the degree of the polynomial.
# It is good practice to make such variable definitions near the top
of your scripts.
# For example, maybe you want to define some plotting options here.
# I'll leave it up to you to figure out what format the variables xlim
and ylim should be in (web searching is your friend).
xlim = [-1,5]
ylim = [-1,5]

# The training data are in file "Yuma_train.csv". This is our training
data containing information
# about various UXO objects that have already been dug up. There are
three columns.
# The first two columns are the data "features" that are collected by
the geophysical instruments for each object.
# For now, don't worry about what these features are, just treat them
as some measurements
# that relate to some characteristics of the buried objects.
# The third column are the labels: each of these UXO objects has been
dug up and identified
# as being one of the five types of munitions known to exist at the
firing range where this data was collected.
# Load the data into a variable named "uxo_train" and do a little
investigation:
# How many rows are there? How many columns are there? What are the
```

```
column names?
# You don't have to provide me with answers to those questions, just
do it for your own practice.

# Load the training data file:
uxo_train = pd.read_csv("Yuma_train.csv")

# Print information about the data:
print(uxo_train)
print("Number of rows in the Yuma_train dataset: ",len(uxo_train))
print("Number of columns in the Yuma_train dataset:
",len(uxo_train.columns))
#The column names are - k1,k2,item
```

```
          k1        k2   item
0   2.026628  1.958092  155mm
1   2.081091  1.505442  155mm
2   2.328653  1.835654  155mm
3   2.407872  1.824523  155mm
4   1.615675  1.037950  105mm
5   1.581017  1.108445  105mm
6   1.541407  1.156678  105mm
7   1.630529  1.227173  105mm
8   1.764212  1.238304  105mm
9   1.932554  1.282827  105mm
10 -0.275695 -0.505512    M75
11 -0.127158 -0.353392    M75
12 -0.127158 -0.338551    M75
13 -0.127158 -0.320000    M75
14 -0.003377 -0.160459    M75
15  0.739307 -0.323710   60mm
16  0.863088 -0.056572   60mm
17  1.194821 -0.030601   60mm
18  0.387770 -0.208693   40mm
19  0.466990 -0.242085   40mm
20  0.585819 -0.112226   40mm
21  0.739307 -0.197562   40mm
Number of rows in the Yuma_train dataset:  22
Number of columns in the Yuma_train dataset:  3
```

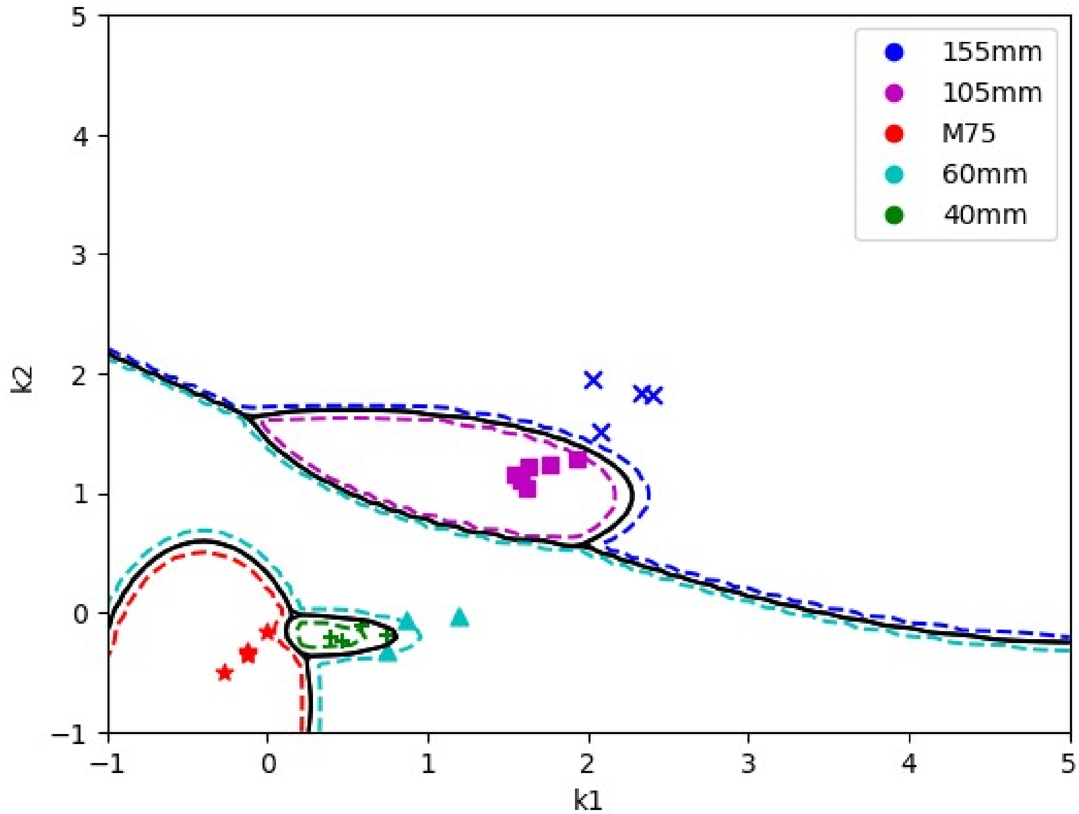
```
# These definitions are required for plotting.
# They define the colours and marker shapes for plotting the different
types of munitions.
# If you printed information about the data above then you should
recognize the names of the munitions here.
# For the colours: b=blue, r=red, c=cyan, m=magenta, g=green,
y=yellow, k=black, and various other options exist.
# For the marker shapes: s=square, ^=triangle, and the others are as
they look.
# For example, a data point for a '155mm' munition should be drawn
```

```
with the colour blue ('b') and as a square ('s') marker.
colors  = {'155mm':'b', '105mm':'m', 'M75':'r', '60mm':'c',
'40mm':'g'}
markers = {'155mm':'x', '105mm':'s', 'M75':'*', '60mm':'^',
'40mm':'+'}

# Create a sensible scatter plot of the data.
# Colour each point differently depending on the label (the type of
munition).
# Add axis and legend annotations to your plot.
# There are various ways to do this but the way I used is to make use
of the pandas "groupby" function,
# which splits the uxo_train variable into different groups, one for
each of the different type of UXO items.
# Create a sensible scatter plot of the data
# Create a sensible scatter plot of the data
groups = uxo_train.groupby('item')
groupScatter(groups, xlim, ylim)

# Axis labels
plt.xlabel('k1')
plt.ylabel('k2')

# Add legend annotations
legend_labels = [plt.Line2D([0], [0], marker='o', color='w',
label=key,
                            markerfacecolor=colors[key], markersize=8)
for key in colors.keys()]
plt.legend(handles=legend_labels, loc='upper right')

# Show the plot
plt.show()
```




```
# The training process should look very similar to what you've done on previous assignments.
# Here you'll use a Gaussian Naive Bayes model. First extract the two features into a variable named X_train
# and extract the labels into a variable named u_train:
# Assuming 'k1' and 'k2' are your features, and 'item' is your label
X_train = uxo_train[['k1', 'k2']]
u_train = uxo_train['item']

# Converting X_train to NumPy array to avoid a warning
X_train = X_train.values


# Here you may need to save some things for later. When you get to the bottom of this Notebook,
# it should become apparent what you need to save here, although the placement of this comment gives it away.
X_train0 = X_train
u_train0 = u_train

model0 = GaussianNB()

# Fit the model using the training data
model0.fit(X_train0, u_train0)
```

```
GaussianNB()

from plotGNB import plotDecisionBoundaries

groups = uxo_train.groupby('item')

#scatter plot
groupScatter(groups, xlim, ylim)

#decision boundaries using the custom function
plotDecisionBoundaries(groups, xlim, ylim, colors, model0)

# Axis labels
plt.xlabel('k1')
plt.ylabel('k2')

# Add legend annotations
legend_labels = [plt.Line2D([0], [0], marker='o', color='w',
label=key,
                            markerfacecolor=colors[key], markersize=8)
for key in colors.keys()]
plt.legend(handles=legend_labels, loc='upper right')

# Show the plot
plt.show()
```



1)The scatter plot visually represents UXO data points with different munition types color-coded in two-dimensional space. The decision boundaries generated by the Gaussian Naive Bayes model separate regions based on classification. The legend associates colors with munition types and axis labels 'k1' and 'k2' indicate the features.

```
# Assume the data features for the unknown item
unknown_features = [[1.0, 0.0]]

# Make a prediction using the trained model
Z = model0.predict_proba(unknown_features)

# Print the predicted probabilities
print("Predicted Probabilities:")
print(Z)

# Print the corresponding UXO item types
print("UXO Item Types:")
print(model0.classes_)

Predicted Probabilities:
[[2.26950808e-48 3.08069991e-37 1.00381150e-05 9.99989962e-01
  3.40375435e-39]]
UXO Item Types:
['105mm' '155mm' '40mm' '60mm' 'M75']
```

2)In the predicted probabilities output, the highest probability 9.99989962e-01(positive value) corresponds to the '60mm' UXO item. This indicates that the model is very confident in predicting the target object as '60mm'. The other probabilities for different UXO item types are close to zero(negative values), suggesting low confidence in those types. Consequently, the most likely type of UXO item for the target object is '60mm', which is consistent with the patterns observed in the plots. The decision to dig up all "40mm" munitions would depend on the predicted probability for '40mm'; if it is also high, it aligns with the assumption to excavate all "40mm" munitions while leaving other types in the ground.

3)In this situation, a "false positive" means our model mistakenly thinks something is a dangerous '40mm' munition when it's not. A "false negative" occurs when the model misses a real '40mm' munition, not recognizing it as dangerous. We're more concerned about reducing false negatives because missing a hazardous item can be risky.

Regarding the object found earlier, if the model predicts a high chance of it being a '40mm' munition, it aligns with our assumption to dig up all dangerous items. However, we should weigh the potential dangers and excavation costs before deciding to dig it up.

```
# Load new data
uxo_new = pd.read_csv("Yuma_new.csv")

# Extract features and labels for new data
X_new = uxo_new[['k1', 'k2']]
u_new = uxo_new['item']

# X_new0 will store the original feature data before any modifications
# u_new0 will store the original feature data before any modifications
X_new0 = X_new.copy()
u_new0 = u_new.copy()

k1 = uxo_new['k1']
k2 = uxo_new['k2']

# Plot new data with decision boundaries
plt.scatter(k1, k2, s=10, color='yellow', label='New Data')
groupScatter(groups, xlim, ylim)
plotDecisionBoundaries(groups, xlim, ylim, colors, model0)

# Prediction for new data
Z = model0.predict_proba(X_new)
print(Z)

# Extract column index for '40mm' item
i40mm = np.where(model0.classes_ == '40mm')[0][0]

# Extract probabilities for '40mm' item
z = Z[:, i40mm]

# Find index of the most likely '40mm' item
imax = np.where(z == max(z))[0].item()
```

```python
# Dig up the most likely '40mm' item
umax = u_new.iloc[uxo_new.index[imax]]
print(umax)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:432:
UserWarning: X has feature names, but GaussianNB was fitted without
feature names
  warnings.warn(
```

```
[[1.03050201e-001 8.96949799e-001 3.69449226e-298 1.80941962e-036
  3.13397285e-158]
 [8.67969658e-006 9.99991320e-001 0.00000000e+000 2.68901284e-043
  6.40187346e-185]
 [3.93923920e-009 9.99999996e-001 0.00000000e+000 4.87323041e-048
  5.68303543e-193]
 [1.12703106e-004 9.99887297e-001 1.30928560e-307 2.43118808e-041
  1.49093643e-195]
 [2.00445341e-006 9.99997996e-001 4.28751452e-317 8.28606947e-044
  1.86018518e-211]
 [5.18126478e-008 9.99999948e-001 0.00000000e+000 1.73815167e-046
  5.47980858e-212]
 [1.76078322e-007 9.99999824e-001 0.00000000e+000 7.40728653e-046
  3.37991343e-200]
 [1.98942467e-008 9.99999980e-001 0.00000000e+000 3.12500021e-047
  2.16339485e-202]
 [3.23401122e-028 1.00000000e+000 0.00000000e+000 2.72321722e-070
  1.47819327e-302]
 [2.34726547e-026 1.00000000e+000 0.00000000e+000 1.36006594e-068
  2.44129975e-284]
 [2.18654049e-024 1.00000000e+000 0.00000000e+000 2.05746622e-066
  1.21175145e-285]
 [1.40685342e-025 1.00000000e+000 0.00000000e+000 8.29115161e-068
  1.83368918e-279]
 [7.41155044e-022 1.00000000e+000 0.00000000e+000 7.88938182e-064
  4.37631694e-267]
 [1.69537813e-021 1.00000000e+000 0.00000000e+000 1.84376228e-063
  4.63483500e-255]
 [8.55566979e-019 1.00000000e+000 0.00000000e+000 1.98632462e-060
  5.13154409e-249]
 [5.62472234e-016 1.00000000e+000 0.00000000e+000 4.41657543e-057
  4.21819875e-231]
 [3.50174070e-013 1.00000000e+000 0.00000000e+000 1.25470645e-053
  2.50155785e-221]
 [4.67996054e-011 1.00000000e+000 0.00000000e+000 7.31069715e-051
  8.24016235e-215]
 [1.19895273e-009 9.99999999e-001 0.00000000e+000 6.39215656e-049
  8.94896650e-214]
 [1.71364399e-010 1.00000000e+000 0.00000000e+000 5.07662448e-050
  7.11974882e-222]
 [5.06460983e-012 1.00000000e+000 0.00000000e+000 4.00387972e-052
```

```
 3.56328747e-224]
[1.12091014e-010 1.00000000e+000 0.00000000e+000 5.61236859e-050
 9.18230079e-233]
[3.87998194e-013 1.00000000e+000 0.00000000e+000 4.09893178e-053
 1.80414615e-247]
[7.02893401e-015 1.00000000e+000 0.00000000e+000 2.05203833e-055
 6.65086964e-253]
[6.92010911e-016 1.00000000e+000 0.00000000e+000 9.62785384e-057
 1.86407268e-254]
[2.36633075e-017 1.00000000e+000 0.00000000e+000 1.52713959e-058
 4.16174047e-259]
[4.74056458e-016 1.00000000e+000 0.00000000e+000 4.17848490e-057
 3.84960050e-248]
[9.25631738e-015 1.00000000e+000 0.00000000e+000 1.28737730e-055
 1.23561900e-234]
[1.22572045e-013 1.00000000e+000 0.00000000e+000 3.47885330e-054
 2.03265204e-233]
[1.09310819e-014 1.00000000e+000 0.00000000e+000 1.84894865e-055
 5.47563876e-241]
[4.68656769e-012 1.00000000e+000 0.00000000e+000 4.32286813e-052
 4.68065541e-230]
[6.20408483e-012 1.00000000e+000 0.00000000e+000 8.00846274e-052
 4.88771177e-234]
[6.61511649e-013 1.00000000e+000 0.00000000e+000 5.56848949e-053
 4.40951601e-242]
[9.84657226e-015 1.00000000e+000 0.00000000e+000 2.31571964e-055
 3.45298638e-248]
[8.00413620e-013 1.00000000e+000 0.00000000e+000 4.88127891e-053
 5.52958266e-236]
[2.06492757e-013 1.00000000e+000 0.00000000e+000 1.05593894e-053
 4.20363999e-242]
[6.17946179e-014 1.00000000e+000 0.00000000e+000 1.86866931e-054
 4.72080947e-241]
[9.94573010e-001 5.42699008e-003 1.58625781e-265 1.35629406e-032
 1.86276879e-142]
[9.99996824e-001 3.17597259e-006 9.45332829e-222 2.93305014e-028
 4.45321626e-129]
[9.99999930e-001 6.98860788e-008 3.26237230e-190 1.74764864e-024
 5.06998506e-120]
[9.99999999e-001 6.84925453e-010 8.65660517e-149 1.83926310e-017
 8.96963807e-097]
[9.99999999e-001 6.11740666e-010 2.85109029e-132 9.01627769e-015
 4.52705927e-093]
[9.99999996e-001 3.52008641e-009 9.61223147e-133 1.20988426e-015
 6.65349767e-103]
[9.99999909e-001 9.11316796e-008 6.70807511e-123 1.18794601e-014
 4.89786629e-115]
[9.99999642e-001 3.58054448e-007 1.29428133e-132 8.83181144e-017
 2.08370242e-125]
```

```
 [9.99999347e-001 6.53071784e-007 8.43082319e-144 6.79503209e-019
  8.27307486e-132]
 [9.99998799e-001 1.20093131e-006 3.60667577e-166 1.51796719e-022
  2.50922690e-138]
 [9.99999013e-001 9.87177380e-007 1.44141502e-155 6.56190022e-021
  2.84803248e-136]
 [9.99999990e-001 9.66166083e-009 7.72749395e-138 6.64816038e-017
  1.07602923e-109]
 [9.99999996e-001 3.52871740e-009 3.97941291e-143 2.00931114e-017
  1.44783500e-105]
 [9.99999984e-001 1.58519534e-008 3.09315001e-147 1.24158630e-018
  2.14441037e-114]
 [9.99999993e-001 7.48249773e-009 1.72220658e-155 1.33015405e-019
  2.51640311e-111]
 [9.99999986e-001 1.39323518e-008 2.73945375e-163 6.53497193e-021
  5.23918620e-115]
 [9.99999980e-001 2.03771536e-008 6.50140324e-158 2.51688110e-020
  5.36102999e-117]
 [9.99999881e-001 1.19361776e-007 3.26465776e-162 1.78021900e-021
  1.42998003e-126]
 [9.99984799e-001 1.52014390e-005 1.41822689e-186 1.09783902e-025
  4.42021188e-150]
 [9.99943126e-001 5.68735355e-005 1.76700852e-179 6.52377425e-025
  7.87240252e-157]
 [1.48181947e-003 9.98518181e-001 6.49444560e-286 5.85395537e-039
  5.44806587e-198]
 [9.20177628e-001 7.98223717e-002 3.16609068e-238 3.94897000e-032
  1.73160393e-177]
 [9.54092226e-001 4.59077737e-002 3.77115047e-226 6.12892436e-031
  1.38954295e-179]
 [9.42689241e-001 5.73107589e-002 5.68727496e-263 1.42330136e-033
  4.59738123e-160]
 [9.92599545e-001 7.40045531e-003 1.16750078e-246 2.76717035e-032
  7.79820723e-160]
 [9.84173166e-001 1.58268336e-002 8.30654033e-240 5.21530988e-032
  4.70942902e-168]
 [9.99858601e-001 1.41399387e-004 1.00605661e-234 1.80490506e-030
  1.57005950e-144]
 [9.99062162e-001 9.37837818e-004 3.19631081e-237 3.27820288e-031
  4.92571787e-154]
 [9.99879179e-001 1.20820698e-004 1.61590881e-223 9.13540879e-030
  5.78099733e-150]
 [9.99985911e-001 1.40893999e-005 2.86514614e-217 1.13558702e-028
  8.99486320e-141]
 [9.99999196e-001 8.03898784e-007 1.89783500e-201 1.82941628e-026
  7.80873789e-131]
 [9.99999554e-001 4.46238519e-007 2.44472363e-191 2.31381334e-025
  1.14627445e-130]
 [9.99999902e-001 9.83903863e-008 1.59876492e-181 8.10273423e-024
```

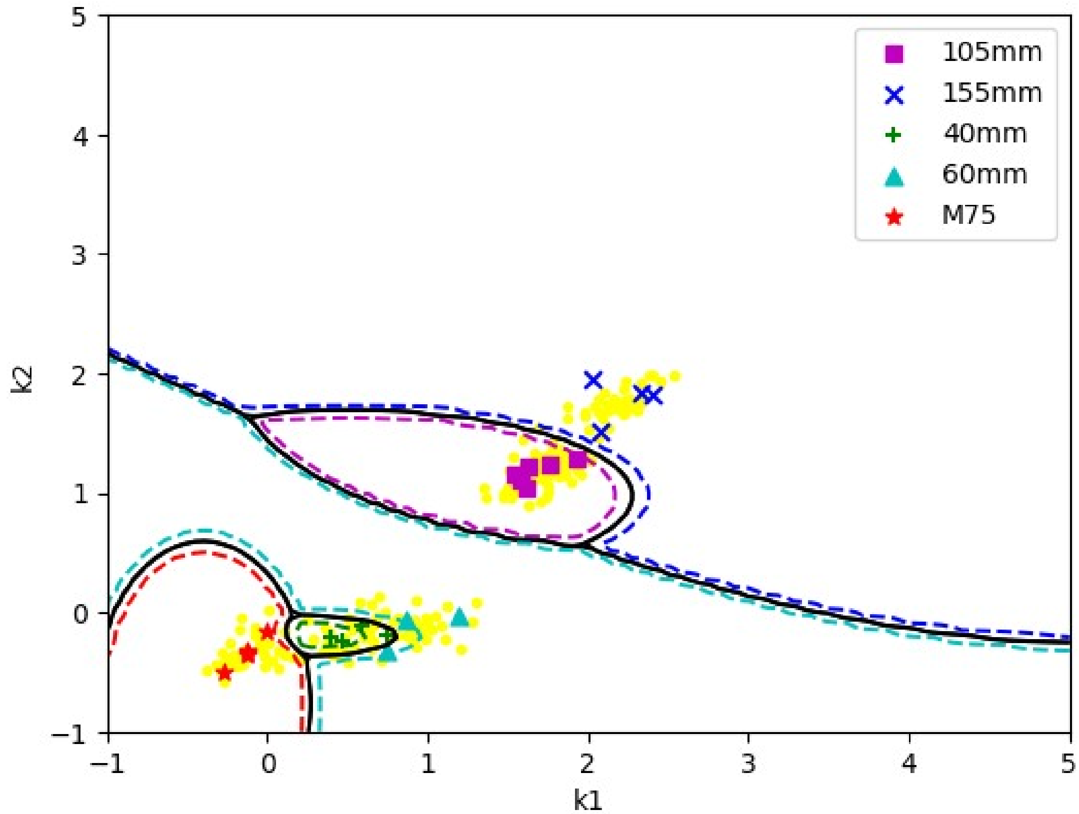
```
  2.91837171e-124]
 [9.99998879e-001 1.12058012e-006 1.52706499e-175 8.13005383e-024
  5.36913876e-138]
 [9.99999734e-001 2.66164217e-007 2.58037233e-175 1.96067234e-023
  1.45312702e-130]
 [9.99999946e-001 5.38677957e-008 1.85482102e-170 2.49515005e-022
  2.85268638e-122]
 [9.99998160e-001 1.83963545e-006 1.85028445e-192 6.44863627e-026
  3.14850040e-138]
 [9.99991323e-001 8.67721494e-006 3.75177591e-208 7.88861853e-028
  5.94671478e-142]
 [9.99998374e-001 1.62611231e-006 3.07815944e-198 1.91282256e-026
  5.41073465e-136]
 [9.99952492e-001 4.75076911e-005 2.40201502e-204 6.94843385e-028
  2.63123827e-152]
 [9.99985924e-001 1.40762924e-005 8.08697354e-203 1.78430938e-027
  2.19579032e-146]
 [9.99972348e-001 2.76517316e-005 1.58247689e-210 2.37806682e-028
  1.94730627e-147]
 [9.99841078e-001 1.58922243e-004 7.11451331e-218 2.16298516e-029
  5.55616250e-154]
 [9.84869462e-001 1.51305382e-002 2.09496654e-248 1.47557799e-032
  9.63240869e-163]
 [9.97423893e-001 2.57610686e-003 4.78410474e-235 2.56098874e-031
  7.36796728e-161]
 [9.99554622e-001 4.45378387e-004 2.88827411e-218 1.25469220e-029
  3.06177183e-159]
 [9.99809877e-001 1.90123254e-004 5.17204426e-206 2.74241944e-028
  7.01766999e-159]
 [9.98193343e-001 1.80665677e-003 1.11218173e-214 1.77424718e-029
  1.58693471e-167]
 [9.92287042e-001 7.71295765e-003 2.86913150e-230 3.91717279e-031
  5.98562314e-169]
 [9.99939347e-001 6.06533300e-005 1.50192015e-192 1.31123208e-026
  4.74333170e-156]
 [9.99987487e-001 1.25134857e-005 2.27902739e-192 2.41668215e-026
  2.92624936e-148]
 [9.99997213e-001 2.78702895e-006 5.15562196e-193 4.37692521e-026
  2.49018751e-140]
 [9.99999117e-001 8.82921453e-007 3.16104904e-181 1.80282204e-024
  2.71578889e-136]
 [9.99996494e-001 3.50560772e-006 2.53864580e-187 1.65281276e-025
  1.30230055e-142]
 [9.99422875e-001 5.77125400e-004 3.04639219e-194 4.94387747e-027
  1.68972004e-166]
 [9.99410680e-001 5.89319735e-004 5.68156223e-203 4.36130698e-028
  3.91018063e-165]
 [9.99904919e-001 9.50807750e-005 1.32756232e-226 6.68704023e-030
  5.09030905e-147]
```

```
[9.99996140e-001 3.85988282e-006 6.84449926e-208 1.50876191e-027
 1.84285136e-137]
[4.91901823e-141 1.77994971e-094 2.06769503e-021 1.13037399e-010
 1.00000000e+000]
[1.35405062e-135 4.24891508e-095 5.09626066e-017 8.83345981e-011
 1.00000000e+000]
[1.25569293e-132 1.83727892e-092 2.51488312e-016 1.57021694e-010
 1.00000000e+000]
[1.55538458e-128 1.09740397e-091 3.39374104e-014 1.86005658e-010
 1.00000000e+000]
[5.19494469e-125 1.05848200e-088 3.99521127e-013 6.28588148e-010
 9.99999999e-001]
[9.95422048e-123 4.09714383e-086 1.98339512e-012 3.44228037e-009
 9.99999997e-001]
[3.75092147e-125 7.74365736e-088 2.25423949e-013 1.21065620e-009
 9.99999999e-001]
[4.17056003e-128 5.89574484e-091 3.47032683e-014 2.38047454e-010
 1.00000000e+000]
[3.81513037e-122 1.64743637e-088 7.62093785e-012 5.37546138e-010
 9.99999999e-001]
[4.62873195e-119 7.33463722e-088 9.15120145e-011 6.29783929e-010
 9.99999999e-001]
[3.92108915e-109 2.55701661e-083 3.23843820e-008 3.78431772e-009
 9.99999964e-001]
[1.44581981e-096 3.28429971e-077 1.58668393e-006 6.83383916e-008
 9.99998345e-001]
[3.93887850e-087 7.10449571e-073 3.64607421e-007 5.72968453e-007
 9.99999062e-001]
[3.45298203e-074 1.34725123e-063 1.81599166e-005 5.65645452e-004
 9.99416195e-001]
[5.19161021e-085 7.35149597e-066 4.17462962e-001 1.98696144e-003
 5.80550076e-001]
[1.16781821e-100 6.24720889e-073 6.64706630e-004 4.34485644e-005
 9.99291845e-001]
[7.66728386e-102 5.42804437e-072 7.51897987e-004 2.91006551e-004
 9.98957095e-001]
[6.19207504e-102 1.36960224e-070 1.49577113e-003 2.78664560e-003
 9.95717583e-001]
[7.92478348e-111 2.01361468e-077 1.72506870e-007 3.20244317e-006
 9.99996625e-001]
[3.52582874e-117 6.52682886e-081 3.10779171e-010 2.59492664e-007
 9.99999740e-001]
[1.56423626e-110 1.30460162e-078 1.67922317e-007 5.91375425e-007
 9.99999241e-001]
[1.55414639e-105 1.99693944e-077 4.77914233e-006 5.33875723e-007
 9.99994687e-001]
[1.90872415e-102 1.07707615e-076 1.82985842e-005 5.00254143e-007
 9.99981201e-001]
[4.01234177e-098 1.13587095e-073 4.86625670e-004 5.77639564e-006
```

```
 9.99507598e-001]
[1.46408186e-099 5.46163790e-075 1.17206566e-004 1.67029376e-006
 9.99881123e-001]
[2.46754578e-096 2.23886412e-074 1.39709156e-004 1.25437868e-006
 9.99859036e-001]
[8.21416614e-096 9.22958894e-075 6.26527049e-005 6.74657821e-007
 9.99936673e-001]
[8.22481588e-094 5.84457440e-074 7.51431416e-005 9.50927736e-007
 9.99923906e-001]
[7.79487382e-091 9.73445950e-073 7.10934362e-005 1.63052665e-006
 9.99927276e-001]
[9.84289359e-110 2.16134601e-081 9.01225536e-008 1.95886313e-008
 9.99999890e-001]
[2.69276000e-112 3.56132426e-082 1.80687216e-008 1.60415218e-008
 9.99999966e-001]
[2.61466810e-118 8.32436897e-085 1.44207875e-010 4.69017471e-009
 9.99999995e-001]
[5.50828381e-121 5.92442658e-087 1.58708125e-011 1.25945271e-009
 9.99999999e-001]
[4.16832324e-117 1.18866495e-085 3.87023027e-010 1.91123971e-009
 9.99999998e-001]
[6.16560822e-043 4.63599109e-036 1.27252156e-008 9.99999987e-001
 5.56945657e-035]
[4.97632742e-043 2.56324878e-033 1.08751249e-008 9.99999989e-001
 5.87784474e-048]
[5.30155967e-038 8.69472863e-029 2.13681584e-012 1.00000000e+000
 2.94221560e-061]
[1.31528654e-049 8.90271099e-034 3.33256204e-007 9.99999667e-001
 2.11008157e-058]
[1.48133486e-050 1.73184815e-037 4.49506040e-005 9.99955049e-001
 9.03021196e-042]
[5.30622278e-056 6.41848338e-039 5.04753489e-004 9.99495247e-001
 3.35868587e-044]
[1.60625634e-070 3.42332945e-042 6.97099814e-006 9.99993029e-001
 3.56847926e-051]
[6.56537791e-065 8.82218425e-042 9.05862126e-004 9.99094138e-001
 7.93054377e-045]
[4.42666723e-088 1.95803422e-055 1.60228802e-003 9.98397712e-001
 1.70007201e-019]
[2.45653078e-081 1.53375127e-054 5.20384444e-001 4.79615556e-001
 2.89459229e-017]
[3.19375544e-077 6.83479087e-057 9.88120089e-001 1.18799110e-002
 5.78613076e-012]
[2.50848237e-059 2.29887605e-046 1.81112870e-001 8.18887130e-001
 1.04828670e-019]
[3.12130613e-068 1.06717776e-048 7.21172615e-001 2.78827385e-001
 1.36042900e-023]
[8.12210926e-068 1.44265628e-047 5.10021814e-001 4.89978186e-001
 2.89876850e-026]
```

```
[1.15232860e-058 1.95840214e-044 9.86842625e-002 9.01315738e-001
 1.89190855e-025]
[1.35943460e-055 1.00113243e-042 1.27634044e-002 9.87236596e-001
 1.39455140e-027]
[7.27518112e-055 5.14937417e-041 3.91658384e-003 9.96083416e-001
 2.71252226e-033]
[3.70968860e-052 5.76029593e-039 2.99434082e-004 9.99700566e-001
 8.48265182e-038]
[8.51522907e-060 2.95325387e-041 6.04432179e-003 9.93955678e-001
 1.34479263e-039]
[8.46973633e-056 3.21048394e-040 2.48138552e-003 9.97518614e-001
 4.67805588e-038]
[9.96922460e-070 1.56497751e-046 9.27355628e-002 9.07264437e-001
 1.74589691e-031]
[9.24059988e-073 3.94313699e-047 2.16186937e-002 9.78381306e-001
 9.78836927e-033]
[8.71831694e-067 3.46764090e-046 2.50492475e-001 7.49507525e-001
 1.78685498e-029]
[1.25444164e-061 6.22337687e-044 9.96273863e-002 9.00372614e-001
 3.02124715e-031]
[5.77794295e-058 7.86553668e-042 1.48928478e-002 9.85107152e-001
 1.75251972e-034]
[2.67756692e-063 2.11906798e-043 3.56024936e-002 9.64397506e-001
 1.50045785e-035]
[2.79749500e-066 4.68296733e-044 1.98728120e-002 9.80127188e-001
 7.24338064e-037]
[8.22329075e-067 1.79917655e-045 1.14963875e-001 8.85036125e-001
 5.29452084e-032]
[1.20241935e-115 1.00304768e-077 1.15468363e-009 1.17870993e-005
 9.99988212e-001]
[3.93619047e-100 5.39256351e-068 1.67717934e-002 8.83755868e-002
 8.94852620e-001]
[7.58780167e-090 8.79482599e-063 9.35061486e-001 6.49136150e-002
 2.48994836e-005]
[2.06518456e-071 9.96124830e-058 9.13534355e-001 8.56176972e-002
 8.47947377e-004]
[8.80292324e-051 4.59876971e-044 1.22177135e-005 9.99987782e-001
 1.36419776e-016]
[6.18925662e-045 6.48035705e-040 1.56552603e-008 9.99999984e-001
 8.93895098e-023]
[1.46555521e-053 4.10548989e-043 2.33654528e-003 9.97663455e-001
 2.20029326e-023]
[1.28510662e-056 4.79835273e-045 2.97984665e-002 9.70201534e-001
 1.27494038e-020]
[5.34979231e-063 2.63731295e-050 5.49991353e-001 4.50008647e-001
 1.70279468e-012]
[2.59677281e-074 1.19172193e-057 9.87417560e-001 1.25823702e-002
 6.96397925e-008]
[1.57337141e-084 9.20094253e-063 9.95454427e-001 4.49776635e-003
```

```
  4.78066432e-005]
 [7.15476088e-091 2.20681817e-061 5.36944238e-001 4.63054879e-001
  8.83386883e-007]
 [2.58455324e-093 2.47457045e-060 1.06295572e-002 9.89370442e-001
  5.79469692e-010]
 [3.19304247e-083 2.21364018e-057 8.84226451e-001 1.15773549e-001
  1.04791117e-012]
 [1.12659935e-090 9.62513054e-061 3.74169892e-001 6.25830015e-001
  9.33160510e-008]
 [8.75625297e-086 2.83877886e-059 8.98429851e-001 1.01570148e-001
  5.30925966e-010]
 [1.09587521e-087 3.91302334e-061 9.35689433e-001 6.43104006e-002
  1.66573928e-007]
 [1.53451442e-084 3.24247799e-061 9.90269662e-001 9.73011056e-003
  2.27548076e-007]
 [1.53469819e-081 1.00687830e-059 9.92174958e-001 7.82503533e-003
  6.96478142e-009]
 [1.21734784e-076 2.94971152e-058 9.92404205e-001 7.59578532e-003
  9.29053645e-009]
 [4.45676483e-076 5.79228281e-053 8.65380485e-001 1.34619515e-001
  6.77445908e-019]
 [1.81368058e-073 1.12907229e-052 9.35679154e-001 6.43208460e-002
  2.95600809e-018]
 [1.13176251e-069 2.71924723e-051 9.22714910e-001 7.72850897e-002
  1.80327895e-018]
 [3.83460670e-065 4.61292971e-049 7.79369549e-001 2.20630451e-001
  2.68203587e-019]
 [2.09102839e-064 1.69575889e-049 7.60263431e-001 2.39736569e-001
  4.77451153e-017]
 [3.63908839e-063 1.44897606e-049 6.35286154e-001 3.64713846e-001
  5.06922993e-015]
 [2.38521194e-071 8.23600066e-054 9.71055249e-001 2.89447513e-002
  5.18061432e-014]
 [2.93007212e-066 2.03639712e-050 8.65136303e-001 1.34863697e-001
  6.31656983e-017]
 [3.00624026e-074 1.08547200e-053 9.60499080e-001 3.95009203e-002
  1.29714175e-016]
 [1.37084867e-066 6.70136765e-052 8.84984928e-001 1.15015072e-001
  3.48245325e-013]
 [6.86669468e-050 8.20066077e-043 1.19022640e-005 9.99988098e-001
  1.50375463e-019]]
40mm
```

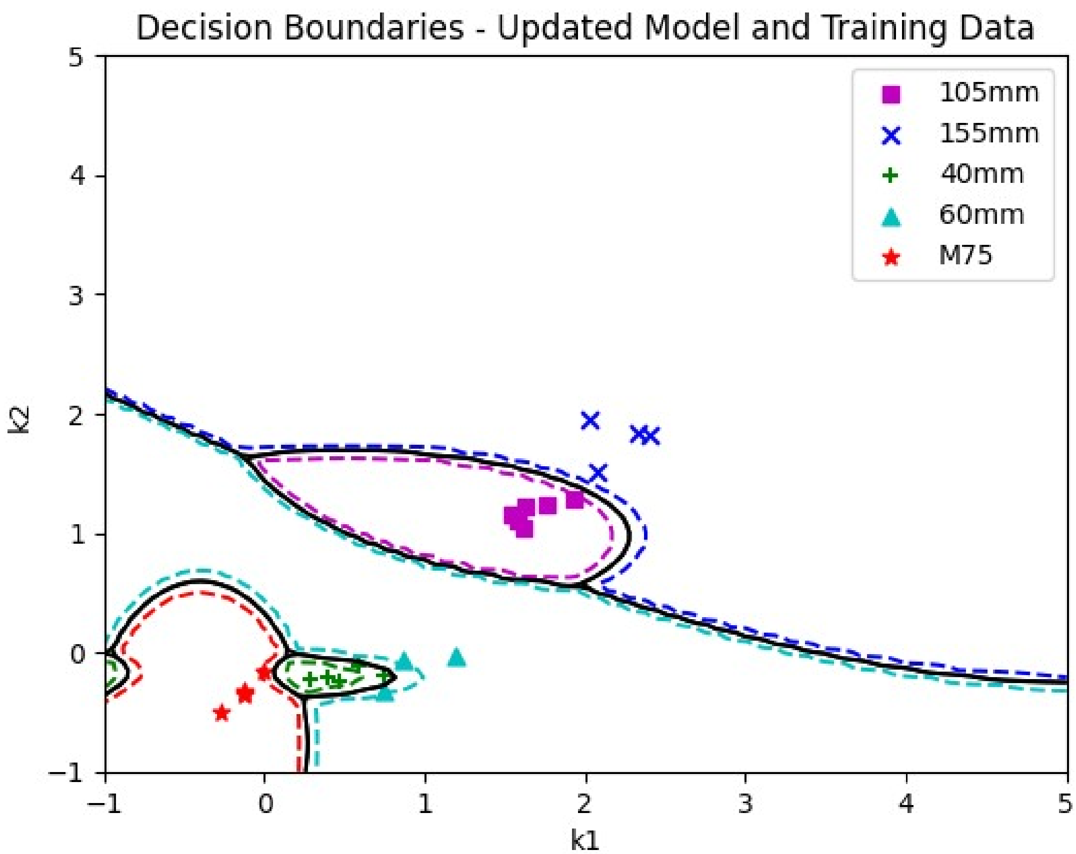


4)The 40mm is the item that was dug up first

```
# Add newly dug item to training set
uxo_add = uxo_new.iloc[[imax]]
uxo_train = pd.concat([uxo_train, uxo_add])
uxo_new = uxo_new.drop(index=uxo_new.index[imax])

# Print number of rows in each variable
print("Number of rows in uxo_train:", uxo_train.shape[0])
print("Number of rows in uxo_new:", uxo_new.shape[0])

# Extract features and labels for updated training set
X_train = uxo_train[['k1', 'k2']]
u_train = uxo_train['item']

# Save for later
X_train0 = X_train.copy()
u_train0 = u_train.copy()

# Create and fit Gaussian Naive Bayes model on updated training set
model1 = GaussianNB()
model1.fit(X_train, u_train)

# Plot decision boundaries with updated model and training data
```

```
groups = uxo_train.groupby('item')
groupScatter(groups, xlim, ylim)
plotDecisionBoundaries(groups, xlim, ylim, colors, model1)
plt.title('Decision Boundaries - Updated Model and Training Data')

# Display the plot
plt.show()
```

```
Number of rows in uxo_train: 23
Number of rows in uxo_new: 187

/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439:
UserWarning: X does not have valid feature names, but GaussianNB was
fitted with feature names
  warnings.warn(
```



5)By comparing the 2 plots I feel the the classification has improved as it include all the data points while plotting and the there can be seen that some overlapping in points at some place but i feel that when we take all the data points in the dataset(including outliers) this could happen(But not really sure)

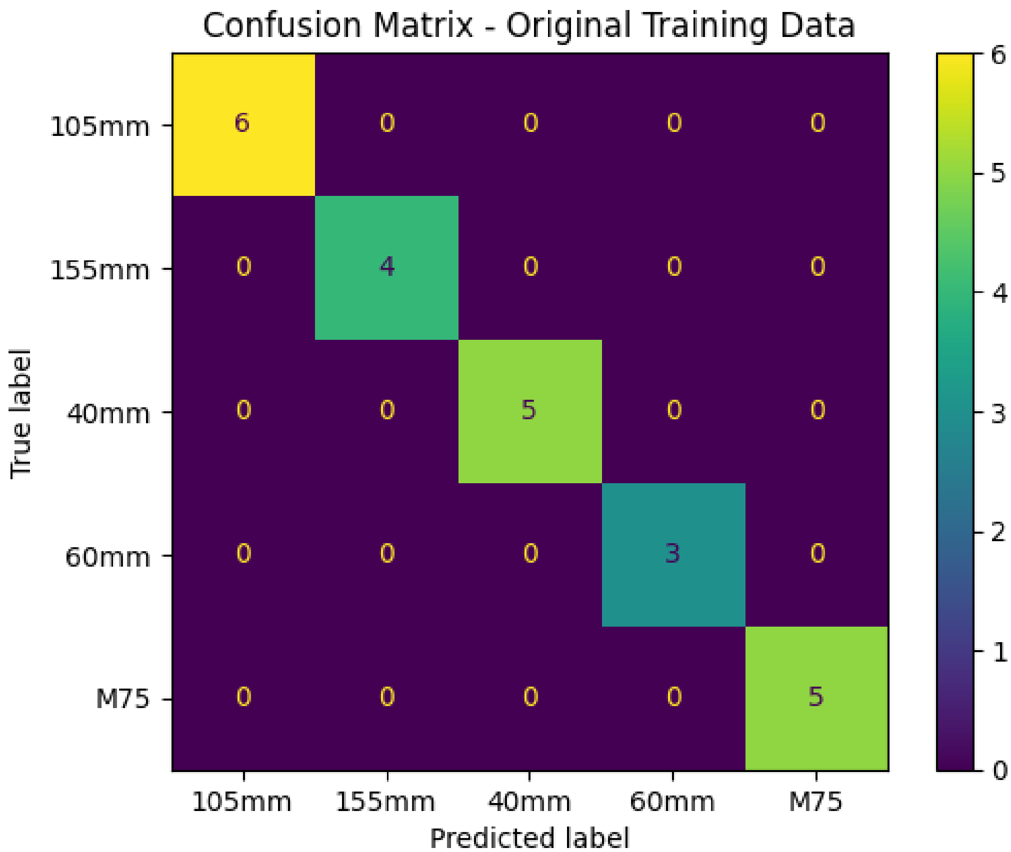
6)The video shows a detector finding the 40mm munitions and re-drawing the decision boundary fro each iteration based on the finding. This makes the machine to learn from the data collected and make necessary changes to find the 40mm munitions. It's trying hard to dig up all the 40mm things without making too many mistakes.

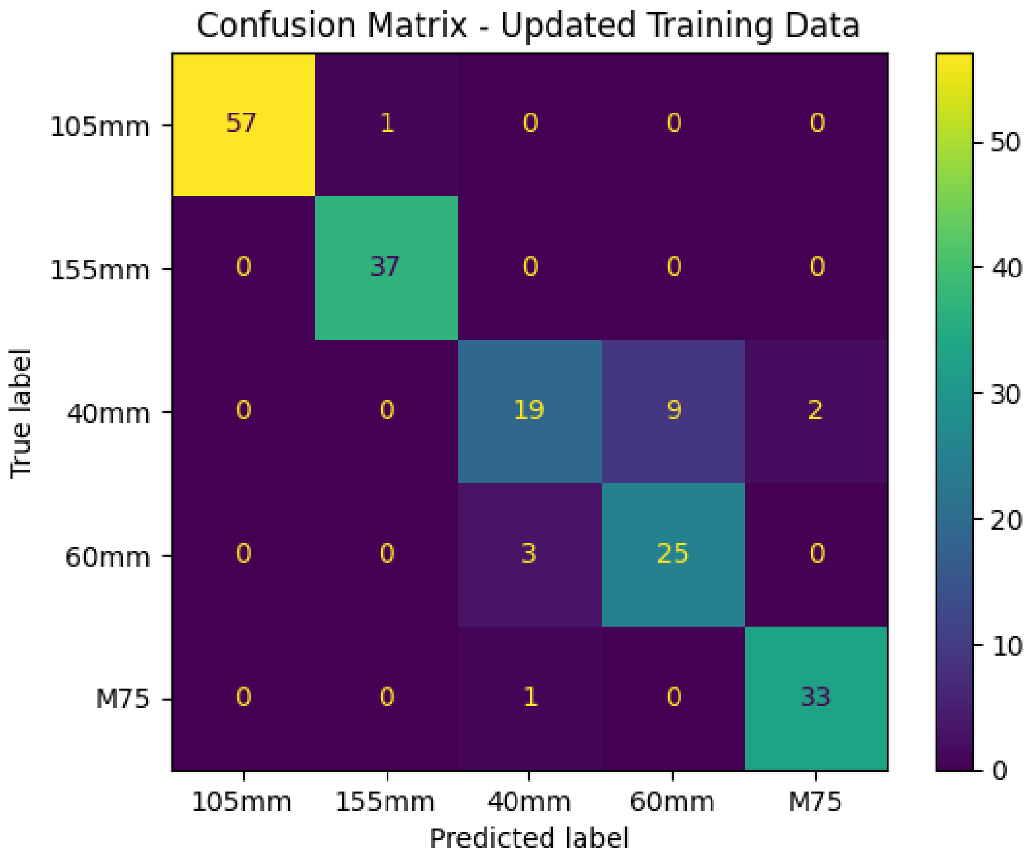
7)the ROC curve for the Gaussian Naive Bayes (GNB) thing has this weird step thing. At first, it is flucuating at different thresholds, and then it sort of levels off on a plateau where the true positive thing stays high and false positives rarely come. There is this little vertical line thing that pops up and that means some threshold stuff happened and false positives shot up suddenly. (not really sure if this is the answer that is being expected)

8)Adjusting classifier parameters, specifically the decision making variable, can improve our chances of finding all 40mm items during field procedures. By tuning the decision making variable to increase sensitivity, we prioritize identifying these items, even if it leads to more false positives. This adjustment is reflected in the ROC curve, where lowering the decision making variables results in a higher true positive rate and an accepted increase in false positives. This iterative approach allows for the betterment of the classifier's performance while emphasizing the goal of minimizing false negatives.

```
# Confusion matrices
X_train = uxo_train[['k1', 'k2']]
u_train = uxo_train['item']

ConfusionMatrixDisplay.from_estimator(model0, X_train0, u_train0)
plt.title('Confusion Matrix - Original Training Data')
plt.show()

X_new = uxo_new[['k1', 'k2']]
u_new = uxo_new['item']
ConfusionMatrixDisplay.from_estimator(model1, X_new, u_new)
plt.title('Confusion Matrix - Updated Training Data')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:432:
UserWarning: X has feature names, but GaussianNB was fitted without
feature names
  warnings.warn(
```


Confusion Matrix - Original Training Data
True label
Predicted label
105mm
155mm
40mm
60mm
M75
6
4
5
3
5
0
1
2
3
4
5
6



9)The updated training data confusion matrix shows more false positives and false negatives than the original matrix it suggests the model might not have gotten better with the new info at this point. This can happen if the added data creates issues at that certain period. And I personally don't think this type of scenario will continue as that the effect of new data on the model's performance varies, sometimes it improves accuracy while other times it may worsen it.

10)More true negatives in the final confusion matrix means the model got better at spotting things that aren't 40mm items. This happened because we removed all 40mm items, letting the model focus on other types. The drop in true positives is likely because we already dug up all 40mm items, giving the model fewer chances to get them right.