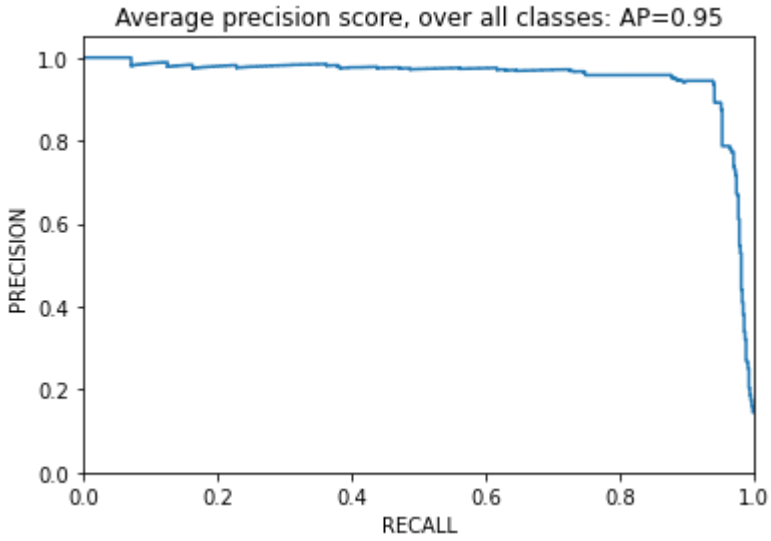
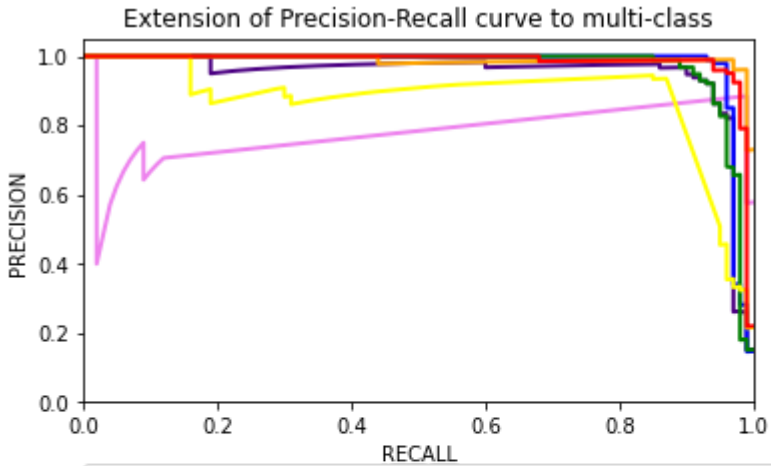
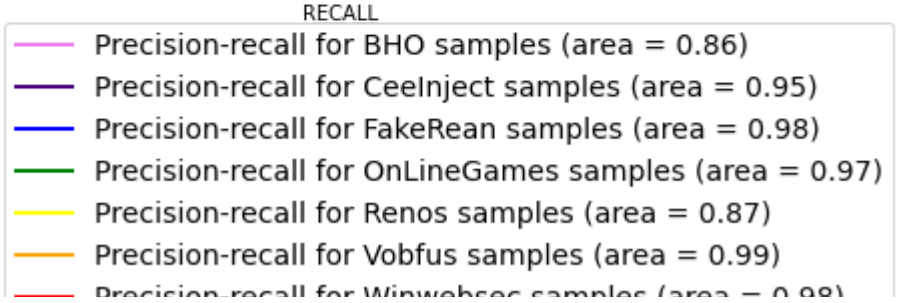
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Mon May 18 11:29:26 2020

@author: yash
"""

import pandas as pd
from sklearn.preprocessing import label_binarize
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.multiclass import OneVsRestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
from itertools import cycle
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import average_precision_score, classification_report
import matplotlib.pyplot as plt
dataset = pd.read_csv('labelledfeatures.csv')
X = dataset.iloc[:,:-2].values
y = dataset.iloc[:,328].values

y = label_binarize(y, classes=[0, 1, 2, 3, 4, 5, 6])

X_train = X[0:5600]
X_test = X[5600:6301]
y_train = y[0:5600]
y_test = y[5600:6301]

# Feature Scaling
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

model = LogisticRegression(solver = 'newton-cg', C = 1, random_state = 42, max_iter = 500)

ovr = OneVsRestClassifier(model)

ovr.fit(X_train, y_train)

y_pred = ovr.predict(X_test)

acc = accuracy_score(y_test, y_pred)
# conf_matrix = confusion_matrix(y_test, y_pred)
print("Accuracy of the model is:")
print(acc)
# print("The confusion matrix is:")
# print(conf_matrix)
print(classification_report(y_test, y_pred, target_names=['BHO', 'CeeInject','FakeRean', 'OnL
y_score = ovr.decision_function(X_test)
```

```
n_classes = 7
precision = dict()
recall = dict()
average_precision = dict()
for i in range(n_classes):
    precision[i], recall[i], _ = precision_recall_curve(y_test[:, i], y_score[:, i])
    average_precision[i] = average_precision_score(y_test[:, i], y_score[:, i])

# A "micro-average": quantifying score on all classes jointly
precision["micro"], recall["micro"], _ = precision_recall_curve(y_test.ravel(), y_score.ravel
average_precision["micro"] = average_precision_score(y_test, y_score, average="micro")

#print('Average precision score over all classes: {0:0.2f}'.format(average_precision["micro"]
plt.figure()
plt.step(recall['micro'], precision['micro'], where='post')

plt.xlabel('RECALL')
plt.ylabel('PRECISION')
plt.ylim([0.0, 1.05])
plt.xlim([0.0, 1.0])
plt.title('Average precision score, over all classes: AP={0:0.2f}'.format(average_precision["
plt.show()

# setup plot details
colors = cycle(['violet', 'indigo', 'blue','green', 'yellow', 'orange', 'red'])
lines = []
labels = []
for i, color in zip(range(n_classes), colors):
    tgt = ['BHO', 'CeeInject','FakeRean', 'OnLineGames','Renos', 'Vobfus','Winwebsec']
    l, = plt.plot(recall[i], precision[i], color=color, lw=2)
    lines.append(l)
    labels.append('Precision-recall for {0} samples (area = {1:0.2f})'''.format(tgt[i], avera

fig = plt.gcf()
fig.subplots_adjust(bottom=0.25)
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('RECALL')
plt.ylabel('PRECISION')
plt.title('Extension of Precision-Recall curve to multi-class')
plt.legend(lines, labels, loc=(0, -1.0), prop=dict(size=14))

plt.show()
```

```
Accuracy of the model is:
0.9257142857142857
              precision    recall  f1-score   support

         BHO       0.88      0.99      0.93       100
   CeeInject       0.90      0.94      0.92       100
    FakeRean       0.98      0.96      0.97       100
 OnLineGames       0.92      0.93      0.93       100
       Renos       0.94      0.85      0.89       100
      Vobfus       0.98      0.97      0.97       100
   Winwebsec       0.98      0.94      0.96       100

   micro avg       0.94      0.94      0.94       700
   macro avg       0.94      0.94      0.94       700
weighted avg       0.94      0.94      0.94       700
 samples avg       0.93      0.94      0.94       700

/usr/local/lib/python3.6/dist-packages/sklearn/metrics/_classification.py:1272: Undefine
  _warn_prf(average, modifier, msg_start, len(result))
```








Precision-Recall for WiFiWebSec samples (area = 0.98)