

```

#!/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
from sklearn.metrics import roc_curve, auc
import numpy as np
from scipy import interp
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

fpr = dict()
tpr = dict()
roc_auc = dict()
for i in range(n_classes):
    fpr[i], tpr[i], _ = roc_curve(y_test[:, i], y_score[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])

# Compute micro-average ROC curve and ROC area
fpr["micro"], tpr["micro"], _ = roc_curve(y_test.ravel(), y_score.ravel())
roc_auc["micro"] = auc(fpr["micro"], tpr["micro"])

plt.figure()
lw = 2
plt.plot(fpr[2], tpr[2], color='darkorange',
         lw=lw, label='ROC curve (area = %0.2f)' % roc_auc[2])
plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic example')
plt.legend(loc="lower right")
plt.show()

# First aggregate all false positive rates
all_fpr = np.unique(np.concatenate([fpr[i] for i in range(7)]))

# Then interpolate all ROC curves at this points
mean_tpr = np.zeros_like(all_fpr)
for i in range(n_classes):
    mean_tpr += interp(all_fpr, fpr[i], tpr[i])

# Finally average it and compute AUC
mean_tpr /= n_classes

fpr["macro"] = all_fpr
tpr["macro"] = mean_tpr
roc_auc["macro"] = auc(fpr["macro"], tpr["macro"])

# Plot all ROC curves
plt.figure()
plt.plot(fpr["micro"], tpr["micro"],
         label='micro-average ROC curve (area = {0:0.2f})'
         ''.format(roc_auc["micro"]),
         color='deeppink', linestyle=':', linewidth=4)

```

```

plt.plot(fpr["macro"], tpr["macro"],
         label='macro-average ROC curve (area = {0:0.2f})'
         ''.format(roc_auc["macro"]),
         color='navy', linestyle=':', linewidth=4)

colors = cycle(['violet', 'indigo', 'blue', 'green', 'yellow', 'orange', 'red'])
for i, color in zip(range(n_classes), colors):
    plt.plot(fpr[i], tpr[i], color=color, lw=lw,
             label='ROC curve of class {0} (area = {1:0.2f})'
             ''.format(i, roc_auc[i]))

plt.plot([0, 1], [0, 1], 'k--', lw=lw)
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Some extension of Receiver operating characteristic to multi-class')
plt.legend(loc=(0, -1.0))
plt.show()

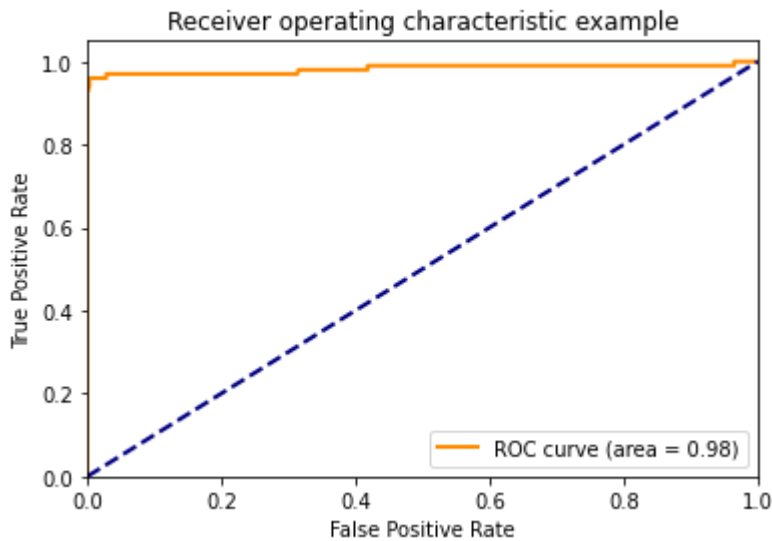
```



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: Undefined
_warn_prf(average, modifier, msg_start, len(result))



/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:87: DeprecationWarning: sci
Some extension of Receiver operating characteristic to multi-class

