

CN240 Data Science for Signal Processing

Code Explanation: Machine Learning

1. Chonlasit Mooncorn 6110613020
2. Prach Chantasantitam 6110613202
3. Weeraphat Leelawittayanon 6210612823
4. Raned Chuphueak 6210612864

Import Library ที่ใช้สำหรับการรับ feature จากไฟล์ dataset ที่ extract มาจาก csv

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import cv2
import os
import imutils
import math
from matplotlib.colors import ListedColormap
```

Input data จากไฟล์ csv เข้ามา จาก feature นั้น ๆ ดังรูป เช่น (diameter และ radius)

```
df = pd.read_csv("idk_final.csv")
g_features = []
g_target = []
count = 0
count_g = 0
count_o = 0

for i in range(len(df["diameter"])):

    #g_features.append([df["diameter"][i],df["area"][i],df["radius"][i]])
    if df["status"][i] == 0:
        if count_g < 842 :
            g_target.append(1)
            g_features.append([df["diameter"][i],df["radius"][i]])
            count_g = count_g + 1

    elif df["status"][i] == 1 :
        if count < 650 :
            g_target.append(0)
            g_features.append([df["diameter"][i],df["radius"][i]])
            count = count + 1

    elif df["status"][i] == 2 :
        if count_o < 150 :
            g_target.append(0)
            g_features.append([df["diameter"][i],df["radius"][i]])
            count_o = count_o + 1

print(f'Select: {len(g_features)} set')
g_features = np.array(g_features)
g_target = np.array(g_target)
```

Import Library ต่าง ๆ มาเพื่อใช้ทำการ cross-validation และ plot กราฟ ROC

```
#cross-validation and plot ROC curves
print(__doc__)

import matplotlib.pyplot as plt
import pickle

from sklearn import svm
from sklearn.metrics import log_loss
from sklearn.metrics import accuracy_score
from sklearn.metrics import auc
from sklearn.metrics import plot_roc_curve
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split
from sklearn.metrics import plot_confusion_matrix
from sklearn.model_selection import StratifiedKFold

# Import some data
```

Import Data เข้ามา และทำการ split ข้อมูลออกมาเป็นจำนวน 5 fold

```
# Import some data
X = g_features
y = g_target

# Add noisy features
random_state = np.random.RandomState(0)

# Train Test split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=random_state)

# Run classifier with cross-validation and plot ROC curves
cv = StratifiedKFold(n_splits=5,shuffle=True)

classifier = svm.SVC(probability=True)
history = []

tprs = []
aucs = []
mean_fpr = np.linspace(0, 1, 100)
fig, ax = plt.subplots()
```

ทำการรันลูป 5 fold ออกมา จัดทำ data และ plot ROC Curve ออกมาเป็น output พร้อมกับตัว model ที่เซฟไว้ใน โฟลเดอร์ดังภาพ

```
for i, (train, val) in enumerate(cv.split(X_train, y_train)):

    X_train, X_val = X[train], X[val]
    y_train, y_val = y[train], y[val]

    # Fit model
    classifier.fit(X_train, y_train)

    # Save model
    with open('Models\\glaucoma_models\\svm_glaucoma', 'wb') as f:
        pickle.dump(classifier, f)

    # Predict
    predict = classifier.predict(X_val)
    clf_probs = classifier.predict_proba(X_val)

    # Store model report in history list
    history.append(classification_report(y_val, predict))

    # Get loss and accuracy
    acc = accuracy_score(y_val, predict)
    loss = log_loss(y_val, clf_probs)
    print(f'=====Fold {i+1}=====','\n')
    print(f"accuracy_score : {acc}")
    print(f"log_loss : {loss}\n")

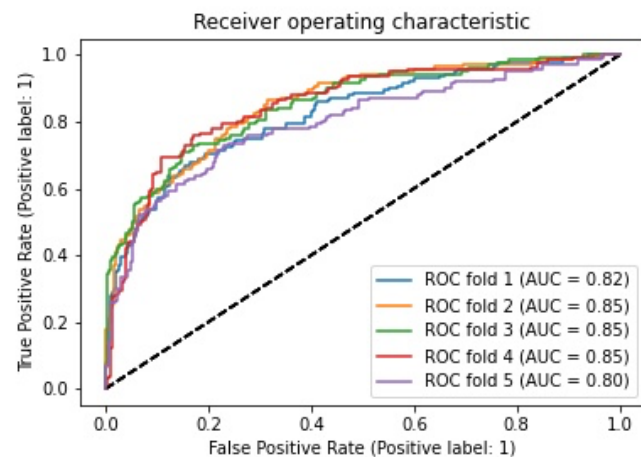
    viz = plot_roc_curve(classifier, X[val], y[val],
                        name='ROC fold {}'.format(i),
                        alpha=0.3, lw=1, ax=ax)
    interp_tpr = np.interp(mean_fpr, viz.fpr, viz.tpr)
    interp_tpr[0] = 0.0
    tprs.append(interp_tpr)
    auks.append(viz.roc_auc)
```

แสดงค่าต่าง ๆ ที่ predict ออกมาที่ output พร้อมกับเซฟรูปภาพกราฟ ROC ตามไฟล์เดอร์ไฟล์ดังภาพ

```
ax.plot([0, 1], [0, 1], 'k--')
ax.set(xlim=[-0.05, 1.05], ylim=[-0.05, 1.05], title="Receiver operating characteristic")
ax.legend(loc="lower right")
plt.savefig('Graph\\glaucoma\\svm_glaucoma_graph.jpg')
plt.show()

predict = classifier.predict(X_test)
print(f"accuracy : {accuracy_score(y_test, predict)}")

for i in range(len(history)):
    print(f'=====Fold {i+1}=====')
    print(history[i])
```



เอาค่าที่ได้จากขั้นตอนที่แล้วมาทำการสร้าง Confusion Matrix ขึ้นแล้วทำการเซฟ

```
from sklearn.metrics import plot_confusion_matrix

# Report
predict = classifier.predict(X_test)
print(classification_report(y_test, predict))
print(accuracy_score(y_test, predict))

# Plot confusion matrix
plot_confusion_matrix(classifier, X_test, y_test, cmap = 'Blues')
plt.savefig('Matrix\\glaucoma\\svm_glaucoma_matrix.jpg')
```

