Bài tập Thực hành môn Khai phá Dữ liệu

Họ và tên: Huỳnh Nguyễn Thế Dân

MSSV: 21110256

Lớp: 21TTH1

Áp dụng với tập dữ liệu banknote authentication từ UCI Machine Learning Repository. Kiểm tra banknote là tuyến tính hay không tuyến tính rồi áp dụng tương tự trường hợp tương ứng như trên.

```
# Import các thư viện cần thiết
import pandas as pd  # Thư viện xử lý dữ liệu
import numpy as np  # Thư viện tính toán khoa học
import matplotlib.pyplot as plt  # Thư viện vẽ biểu đồ
from sklearn.model_selection import train_test_split  # Thư viện chia tập dữ liệu
from sklearn.svm import SVC  # Thư viện hỗ trợ vector machines
from sklearn.metrics import accuracy_score, classification_report  # Thư viện đánh giá mô hình
from sklearn.preprocessing import StandardScaler  # Thư viện chuẩn hóa dữ liệu
from sklearn.decomposition import PCA  # Thư viện phân tích thành phần chính

# Bước 1: Tải dữ liệu
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/00267/data_banknote_authentication.txt"
# Đọc dữ liệu từ URL và đặt tên cho các cột
data = pd.read_csv(url, header=None, names=['Variance', 'Skewness', 'Curtosis', 'Entropy', 'Class'])

# Hiển thị một số hàng đầu tiên của dữ liệu để kiểm tra
print(data.head())
```

Tóm tắt thuật toán:

Đoạn code này thực hiện các bước sau:

1. **Import các thư viện cần thiết**: Sử dụng `pandas` để xử lý dữ liệu, `numpy` để tính toán khoa học, `matplotlib` để vẽ biểu đồ, và các thư viện từ `scikit-learn` để tiền xử lý và xây dựng mô hình máy học.

2. **Tải dữ liệu**:

- Sử dụng `pandas` để tải dữ liệu từ một URL, tập dữ liệu này liên quan đến việc xác thực tiền giấy.
- Đặt tên cho các cột là `Variance`, `Skewness`, `Curtosis`, `Entropy`, và `Class`.
- Hiển thị một vài hàng đầu tiên của tập dữ liệu để kiểm tra.

```
# Bước 2: Kiểm tra tính tuyến tính
# Vẽ biểu đồ phân tán của hai đặc trưng: Variance và Skewness, phân loại theo nhãn Class
plt.scatter(data['Variance'], data['Skewness'], c=data['Class'], cmap='coolwarm')
plt.xlabel('Variance')  # Gán nhãn trục x
plt.ylabel('Skewness')  # Gán nhãn trục y
plt.title('Scatter plot of Variance vs Skewness')  # Đặt tiêu đề cho biểu đồ
plt.show()  # Hiển thị biểu đồ
```

Tóm tắt thuật toán:

- Bước này nhằm kiểm tra tính tuyến tính của dữ liệu bằng cách vẽ biểu đồ phân tán của hai đặc trưng `Variance` và `Skewness`.
- Biểu đồ này giúp ta trực quan hóa mối quan hệ giữa hai đặc trưng và phân loại dữ liệu dựa trên nhãn `Class`.

```
# Bước 3: Áp dụng SVM
# Tách dữ liệu thành tập huấn luyện và tập kiểm tra
X = data[['Variance', 'Skewness', 'Curtosis', 'Entropy']]
y = data['Class']

# Chuẩn hóa dữ liệu
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Sử dụng PCA để giảm chiều dữ liệu xuống còn 2 chiều để trực quan hóa
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

# SVM với kernel RBF
svm_model = SVC(kernel='rbf', C=1E6, gamma='scale')
svm_model.fit(X_pca, y)

# Hàm vẽ biên quyết định
def plot_svc_decision_function(model, ax=None, plot_support=True):
    """Vẽ biên quyết định cho SVM 2D"""
    if ax is None:
        ax = plt.gca()
    xlim = ax.get_xlim()
    ylim = ax.get_ylim()

    # Tạo lưới để đánh giá mô hình
```

```
    x = np.linspace(xlim[0], xlim[1], 30)
    y = np.linspace(ylim[0], ylim[1], 30)
    Y, X = np.meshgrid(y, x)
    xy = np.vstack([X.ravel(), Y.ravel()]).T
    P = model.decision_function(xy).reshape(X.shape)

    # Vẽ biên quyết định và margins
    ax.contour(X, Y, P, colors='k',
               levels=[-1, 0, 1], alpha=0.5,
               linestyles=['--', '-', '--'])

    # # Vẽ các điểm hỗ trợ
    # if plot_support:
    #     ax.scatter(model.support_vectors_[:, 0],
    #                model.support_vectors_[:, 1],
    #                s=300, linewidth=1, edgecolors='black',
    #                facecolors='none')
    # ax.set_xlim(xlim)
    # ax.set_ylim(ylim)

# Vẽ biểu đồ phân tán và biên quyết định
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=y, s=50, cmap='autumn')
plot_svc_decision_function(svm_model)
plt.show()

# Hiển thị các vector hỗ trợ của mô hình
svm_model.support_vectors_
```

Tóm tắt thuật toán:

Đoạn code này tiếp tục từ bước 3, thực hiện các bước sau:

1. **Tách dữ liệu**: Tách dữ liệu thành các đặc trưng (`X`) và nhãn (`y`).
2. **Chuẩn hóa dữ liệu**: Sử dụng `StandardScaler` để chuẩn hóa dữ liệu.
3. **Giảm chiều dữ liệu**: Sử dụng `PCA` để giảm số chiều dữ liệu xuống còn 2 để trực quan hóa.
4. **Áp dụng SVM**: Sử dụng mô hình SVM với kernel RBF để huấn luyện dữ liệu đã giảm chiều.
5. **Vẽ biên quyết định**: Vẽ biên quyết định của SVM trên dữ liệu đã giảm chiều.

Hết.