THỰC HÀNH 6: GIẢM CHIỀU VÀ PHÂN CỤM DỮ LIỆU

Mục tiêu: Đánh giá khả năng sinh viên trong việc áp dụng các kỹ thuật máy học thống kê để giải quyết vấn đề thực tế, cụ thể là trong lĩnh vực giảm chiều dữ liệu và phân cụm dữ liệu phức tạp.

Phần 1: Principal Component Analysis (PCA)

Mục tiêu: Hiểu và áp dụng PCA để giảm chiều dữ liệu và phân tích thành phần chính.

Dữ liệu: Sử dụng bộ dữ liệu Delay Prediction hoặc bất kỳ bộ dữ liệu nhiều chiều nào.

Bài tập cơ bản

1. **Tiền xử lý dữ liệu:** Chuẩn hóa dữ liệu.

Code tham khảo:

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data)
```

2. **Xây dựng và Áp dụng PCA:**

- ***Tạo hàm myPCA và áp dụng lên bộ dữ liệu.***

Code tham khảo:

```
import numpy as np
def myPCA(A):
    cov_matrix = np.cov(A.T)
    eigenvalues, eigenvectors = np.linalg.eig(cov_matrix)
    return eigenvectors, eigenvalues
```

- Áp dụng PCA cho tập dữ liệu Delay Prediction rồi sau đó áp dụng thuật toán K-means trên thành phần chính thứ nhất. Liệu phân bổ cụm có phù hợp khi sử dụng K-means trên tập dữ liệu gốc không? Điều gì xảy ra khi bạn sử dụng thành phần chính thứ hai để thực hiện phân cụm?
- Vẽ các điểm dữ liệu trong không gian 2 chiều thu được từ hai thành phần chính đầu tiên.
- Chọn số lượng thành phần chính và giải thích lựa chọn.

Câu hỏi yêu cầu:

- Giải thích tại sao PCA là một phương pháp hữu ích trong giảm chiều dữ liệu.
- Làm thế nào để quyết định số lượng thành phần chính cần sử dụng?
- So sánh PCA với ít nhất một kỹ thuật giảm chiều dữ liệu khác.

Phần 2: Phân cụm dữ liệu (Clustering)
Mục tiêu: Hiểu và sử dụng thuật toán K-means clustering để phân cụm dữ liệu.

Dữ liệu: Sử dụng bộ dữ liệu Delay Prediction hoặc bất kỳ bộ dữ liệu nhiều chiều nào.

Bài tập cơ bản

1. **Tiền xử lý dữ liệu:** Chuẩn hóa dữ liệu.
2. **Áp dụng K-Mean Clustering:**
    - Tạo một hàm myKmeans sẽ nhận một tập dữ liệu A và các trung tâm cụm ngẫu nhiên ban đầu, và áp dụng thuật toán K-means (có thể tạo một hàm vẽ biểu đồ có thể được gọi trong mỗi lần lặp để vẽ sự phân bổ cụm).

```
def myKmeans(A, num_clusters):
# Implementation of K-means
return clusters, centroids
```

    - Sklearn có một hàm tích hợp sẵn là kmeans, thực hiện phân cụm K-means trên một tập dữ liệu quan sát đã cho. Kiểm tra xem hàm này nhận những tham số nào làm đầu vào và áp dụng nó để phân chia dữ liệu Delay Prediction thành K = 2 cụm. Liệu các trung tâm cụm có phù hợp với cách triển khai của bạn không?
    - Chọn số lượng cụm phù hợp (sử dụng Elbow hoặc phương pháp khác).
    - Áp dụng K-means và phân tích đặc điểm của từng cụm.
    - Tính toán chỉ số Silhouette, Hopkins và phân tích kết quả.
3. **Biểu diễn kết quả:** Trực quan hoá các cụm và giải thích ý nghĩa/

Câu hỏi yêu cầu:

- Giải thích sự cần thiết của việc biến đổi, chuẩn hóa dữ liệu trước khi thực hiện Phân cụm.
- So sánh và đối chiếu giữa kết quả của K-means và Hierarchical clustering.

- Thảo luận về cách lựa chọn số lượng cụm và ảnh hưởng của nó đến kết quả.
- So sánh K-means với một thuật toán phân cụm khác, như DBSCAN hoặc hierarchical clustering.

Lưu ý:

- Sinh viên cần chú trọng đến việc giải thích quy trình phân tích và lựa chọn kỹ thuật.
- Khuyến khích sử dụng ngôn ngữ lập trình Python hoặc R.

Code tham khảo đọc dữ liệu:

```
import glob
import numpy as np
import cv2

IMG_SIZE = 227

def load_dataset(path):
    X = np.array([])
    y = np.array([])
    classes = ['NORMAL', 'PNEUMONIA']
    for c in classes:
        files = glob.glob(path + c + "/*.jpeg")
        for f in files:
            print(f)
            img = cv2.imread(f)
            img = cv2.resize(img, (IMG_SIZE, IMG_SIZE))
            if X.size == 0:
                X = np.array([img])
            else:
                X = np.vstack([X,[img]])
            y = np.append(y, c)

    assert(X.size > 0), 'Cannot read file'
    return (X,y)
```