**BÀI 1: THỰC HÀNH VỚI WEKA**

**Bước 1: Chuẩn bị dữ liệu**

* Mở file cruise\_ship\_info.csv bằng Excel hoặc phần mềm tương đương.
* Xóa các cột dạng **text không cần thiết** cho phân cụm, ví dụ: Ship\_name, Cruise\_line.
* Giữ lại các cột dạng số như:
  + Tonnage
  + passengers
  + length
  + cabins
  + crew
  + passenger\_density
  + year\_built
* Lưu lại dưới dạng **CSV**, sau đó chuyển đổi sang định dạng **ARFF** trong Weka như sau:
  + Weka → Explorer → Open file → chọn file CSV
  + Chọn **Save** để lưu lại dưới dạng .arff (tên: cruise\_ship\_info.arff)

**Bước 2: Thực hiện phân cụm**

* Vào tab Cluster → chọn thuật toán SimpleKMeans
* Thiết lập:
  + numClusters = 3 (hoặc tự chọn số cụm phù hợp)
  + distanceFunction = EuclideanDistance
* Bấm Start để chạy.

**Bước 3: Đánh giá**

* Tick chọn “**Classes to clusters evaluation**” nếu file có nhãn Cruise\_line và bạn giữ lại.
* Quan sát cụm được tạo → Ghi lại:
  + Số lượng mẫu trong mỗi cụm
  + Giá trị trung bình của từng thuộc tính theo từng cụm
  + Nhận xét về mối liên hệ giữa cụm và các loại tàu hoặc hãng tàu (nếu có)

**TRẢ LỜI CÂU HỎI**

* Bạn đã chọn bao nhiêu cụm? Vì sao?
* Mỗi cụm có đặc điểm gì nổi bật (về số hành khách, trọng tải, chiều dài, v.v)?
* Có sự tương đồng giữa cụm và loại tàu/hãng tàu không?
* Nếu thay đổi số cụm thành 4 hoặc 5 thì có cải thiện không?
* So sánh với kết quả phân loại (nếu dữ liệu có nhãn Cruise\_line).

**BÀI 2: THỰC HÀNH VỚI PYTHON**

**Mục tiêu:**

* Hiểu và áp dụng kỹ thuật phân cụm KMeans trong Python.
* Trực quan hóa kết quả phân cụm bằng PCA.
* Phân tích các nhóm tàu du lịch dựa trên đặc trưng kỹ thuật (không dùng nhãn sẵn).

**DATASET**

File cruise\_ship\_info.csv bao gồm thông tin kỹ thuật về các tàu du lịch:

* Các thuộc tính đáng chú ý:
  + Tonnage: tải trọng
  + passengers: số hành khách
  + length: chiều dài tàu
  + cabins: số cabin
  + crew: số thủy thủ
  + passenger\_density: mật độ hành khách
  + year\_built: năm đóng tàu

**Yêu cầu thực hiện:**

Phần 1 – Tiền xử lý

1. Đọc file cruise\_ship\_info.csv bằng pandas.
2. Loại bỏ các cột không mang ý nghĩa phân cụm như Ship\_name, Cruise\_line.
3. Chuẩn hoá dữ liệu với StandardScaler.
4. Kiểm tra và xử lý missing values (nếu có).

Phần 2 – Phân cụm KMeans

1. Áp dụng thuật toán KMeans với số cụm k = 3.
2. Thêm kết quả phân cụm vào DataFrame (cluster).

Phần 3 – Trực quan hóa

1. Giảm chiều dữ liệu xuống 2 thành phần bằng PCA.
2. Vẽ biểu đồ scatterplot các cụm theo trục pca1 và pca2 với màu khác nhau.

Phần 4 – Phân tích

1. Tính giá trị trung bình các thuộc tính kỹ thuật theo từng cụm.
2. Đặt tên gợi ý cho mỗi cụm (ví dụ: "tàu hiện đại", "tàu cũ nhỏ", "tàu hạng sang").

**HINT THỰC HÀNH**

**Bước 1: Import thư viện**

# Gợi ý: pandas, numpy, matplotlib, seaborn, sklearn.cluster, sklearn.preprocessing

import \_\_\_\_\_\_\_\_ as pd

import \_\_\_\_\_\_\_\_ as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.cluster import \_\_\_\_\_\_\_\_

from sklearn.preprocessing import \_\_\_\_\_\_\_\_

from sklearn.decomposition import PCA

**Bước 2: Đọc dữ liệu**

# Gợi ý: Đọc file cruise\_ship\_info.csv

df = pd.\_\_\_\_\_\_\_\_\_\_('cruise\_ship\_info.csv')

df.head()

**Bước 3: Tiền xử lý**

# Gợi ý: Xoá các cột dạng text không cần thiết như Ship\_name, Cruise\_line

df\_clean = df.drop(columns=['\_\_\_\_\_\_\_\_\_\_', '\_\_\_\_\_\_\_\_\_\_'])

# Kiểm tra missing values

print(df\_clean.\_\_\_\_\_\_\_\_\_\_())

# Chuẩn hoá dữ liệu (StandardScaler)

scaler = StandardScaler()

X\_scaled = scaler.\_\_\_\_\_\_\_\_\_\_(df\_clean)

**Bước 4: Áp dụng Elbow Method để chọn k (tùy chọn)**

# Gợi ý: chạy từ k = 1 đến 10 và lưu WCSS

wcss = []

for i in range(1, 11):

kmeans = KMeans(n\_clusters=i, init='k-means++', random\_state=42)

kmeans.fit(\_\_\_\_\_\_\_\_\_\_)

wcss.append(kmeans.\_\_\_\_\_\_\_\_\_\_)

# Vẽ biểu đồ Elbow

plt.plot(range(1, 11), wcss)

plt.title('Phương pháp Elbow')

plt.xlabel('Số cụm (k)')

plt.ylabel('WCSS')

plt.show()

**Bước 5: Áp dụng KMeans clustering**

# Gợi ý: chọn số cụm = 3

kmeans = \_\_\_\_\_\_\_\_\_\_(n\_clusters=3, init='k-means++', random\_state=42)

clusters = kmeans.\_\_\_\_\_\_(X\_scaled)

# Gán nhãn cụm vào bảng gốc

df['cluster'] = clusters

**Bước 6: Giảm chiều PCA (2D) để trực quan**

# Dùng PCA để vẽ biểu đồ 2 chiều

pca = PCA(n\_components=2)

X\_pca = pca.\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_)

df['pca1'] = X\_pca[:, \_\_\_\_\_]

df['pca2'] = X\_pca[:, \_\_\_\_\_]

**Bước 7: Trực quan hoá kết quả phân cụm**

# Vẽ scatterplot theo cụm

sns.\_\_\_\_\_\_\_\_\_(data=df, x='pca1', y='pca2', hue='cluster', palette='Set2')

plt.\_\_\_\_\_\_\_\_('Phân cụm tàu du lịch với PCA')

plt.\_\_\_\_\_\_\_\_('PCA 1')

plt.\_\_\_\_\_\_\_\_\_('PCA 2')

plt.show()

**Bước 8: Phân tích kết quả**

# Gợi ý: Tính trung bình mỗi cụm

group\_summary = df.groupby('cluster').mean(numeric\_only=True)

print(group\_summary)

**TRẢ LỜI CÂU HỎI SAU**

* Mỗi cụm có đặc điểm gì nổi bật (năm sản xuất, tải trọng, số hành khách, v.v)?
* Nếu chọn k = 4 hoặc 5 thì có hợp lý hơn không? Vì sao?
* Có thuộc tính nào gây nhiễu, nên loại bỏ không?
* Tên gợi ý cho các cụm: (gợi ý: “tàu hiện đại – cao cấp”, “tàu nhỏ – cũ”,...)

*Lưu ý: SV thực hiện trên máy và trả lời các câu hỏi. GV chấm kết quả dựa trên kết quả các câu trả lời theo thang điểm từng câu là 5 điềm.*