TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN
KHOA CÔNG NGHỆ THÔNG TIN

--------- oOo ---------

BÀI TẬP THỰC HÀNH

NHẬP MÔN KHAI PHÁ DỮ LIỆU

TRÌNH ĐỘ: ĐẠI HỌC

NGÀNH ĐÀO TẠO: KHOA HỌC MÁY TÍNH

Hưng Yên – Tháng 08 năm 2022

BÀI THỰC HÀNH SỐ 2: PHÂN CỤM DỮ LIỆU VỚI DBSCAN và WEKA

A. MỤC TIÊU BÀI THỰC HÀNH

Sau bài thực hành này sinh viên có thể:

- Hiểu nguyên lý hoạt động của thuật toán DBSCAN dùng trong phân cụm dữ liệu
- Cài đặt được thuật toán DBSCAN cho phân cụm dữ liệu
- Vận dụng kiến thức về DBSCAN để cài đặt ứng dụng khai phá dữ liệu trong thực tế

B. ĐIỀU KIỆN THỰC HÀNH

Với đặc thù của môn Lập trình Python nâng cao, mục này sẽ liệt kê một số công cụ sử dụng để làm bài thực hành. Trong bài thực hành này, sinh viên cần kiểm tra và chắc chắn các phần mềm sau trên máy tính còn hoạt động tốt:

1. Anacoda phiên bản 3.0 trở lên
2. Jupyter Notebook
3. Visual Studio

C. TÀI NGUYÊN THAM CHIẾU

Để hoàn thành tốt bài thực hành này sinh viên nên tham khảo các tài nguyên sau:

STT	Tên tài nguyên	Mô tả tài nguyên
1	Bai tap thuc hanh 02.pdf	Tài liệu bài thực hành số 02
2	Lesson 03 – Phan cum du lieu.pdf	Slide bài giảng về phân cụm dữ liệu

D. YÊU CẦU BÀI THỰC HÀNH

Bài 2.1. Cho dữ liệu có chứa chỉ số chiều cao và cân nặng của 200 người được lưu trong tệp **height_weight.csv**

a) Viết chương trình sử dụng thuật toán Dbscan. Lựa chọn eps, min_sample phân cụm dữ liệu trên thành *k*
cụm (k=2, 3, 4, 5)

b) Trực quan hóa các cụm dữ liệu ứng với các *k*

c) Đánh giá kết quả phân cụm;

- Tính tỉ lệ nhiễu (Noise Ratio) trong bộ dữ liệu
- Dùng chỉ số **Silhouette, DBI** để đánh giá chất lượng cụm.

d) Tìm và lọc các điểm nhiễu và chạy lại DBScan

e) So sánh Kmeans và Dbscan

f) Chuẩn hóa dữ liệu bằng MinMaxScaler và StandardScaler. So sánh ảnh hưởng của việc chuẩn hóa dữ liệu trước và sau khi chuẩn hóa thì ảnh hưởng thế nào đến việc phân cụm

g) Sử dụng WEKA để áp dụng thuật toán phân cụm K-Means

h) Sử dụng WEKA để áp dụng thuật toán DBScan

i) Tìm eps và min_sample tốt nhất để tìm ra giá trị nào cho kết quả phân cụm tốt nhất (số cụm đúng và ít nhiễu nhất).

Bài 2.2. Cho dữ liệu có chứa thông tin về giới tính, chiều cao và cân nặng của 200 người được lưu trong tệp **gender_height_weight.csv**

a) Viết chương trình sử dụng thuật toán Dbscan. Lựa chọn eps, min_sample phân cụm dữ liệu trên thành *k*
cụm (k=2, 3, 4, 5)

b) Trực quan hóa các cụm dữ liệu ứng với các *k*

c) Đánh giá kết quả phân cụm;

- Tính tỉ lệ nhiễu (Noise Ratio) trong bộ dữ liệu
- Dùng chỉ số **Silhouette, DBI** để đánh giá chất lượng cụm.

d) Tìm và lọc các điểm nhiễu và chạy lại DBScan

e) So sánh Kmeans và Dbscan

f) Chuẩn hóa dữ liệu bằng MinMaxScaler và StandardScaler. So sánh ảnh hưởng của việc chuẩn hóa dữ liệu trước và sau khi chuẩn hóa thì ảnh hưởng thế nào đến việc phân cụm

g) Sử dụng WEKA để áp dụng thuật toán phân cụm K-Means

h) Sử dụng WEKA để áp dụng thuật toán DBScan

i) Tìm eps và min_sample tốt nhất để tìm ra giá trị nào cho kết quả phân cụm tốt nhất (số cụm đúng và ít nhiễu nhất).

Bài 2.3. Cho dữ liệu có chứa thông tin về các khách hàng: mã khách hàng, giới tính, tuổi, thu nhập hàng năm và điểm tiết kiệm (1-100) lưu trong tệp **mall_customers.csv**

a) Viết chương trình sử dụng thuật toán Dbscan. Lựa chọn eps, min_sample phân cụm dữ liệu trên thành *k*
cụm (k=2, 3, 4, 5)

b) Trực quan hóa các cụm dữ liệu ứng với các *k*

c) Đánh giá kết quả phân cụm;

- Tính tỉ lệ nhiễu (Noise Ratio) trong bộ dữ liệu
- Dùng chỉ số **Silhouette, DBI** để đánh giá chất lượng cụm.

d) Tìm và lọc các điểm nhiễu và chạy lại DBScan

e) So sánh Kmeans và Dbscan

f) Chuẩn hóa dữ liệu bằng MinMaxScaler và StandardScaler. So sánh ảnh hưởng của việc chuẩn hóa dữ liệu trước và sau khi chuẩn hóa thì ảnh hưởng thế nào đến việc phân cụm

g) Sử dụng WEKA để áp dụng thuật toán phân cụm K-Means

h) Sử dụng WEKA để áp dụng thuật toán DBScan

i) Tìm eps và min_sample tốt nhất để tìm ra giá trị nào cho kết quả phân cụm tốt nhất (số cụm đúng và ít nhiễu nhất).

E. HƯỚNG DẪN THỰC HIỆN

Sinh viên tạo tệp **Practice02_HoVaTen.ipynb** trên Jupyter Notebook và thực hiện viết mã lệnh để giải quyết các bài tập thực hành.

Bài 2.1:

- Sử dụng Pandas để đọc dữ liệu từ file csv.
- Sử dụng thư viện **sklearn.cluster** để phân cụm với K-mean.
- Sử dụng matplotlib (scatter) để trực quan hóa dữ liệu.

Code tham khảm:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

data = pd.read_csv('height_weight.csv')

data.columns=['height','weight']

kmeans = KMeans(n_clusters=3).fit(data)
centroids = kmeans.cluster_centers_
print(centroids)

plt.scatter(data['height'], data['weight'], c= kmeans.labels_.astype(float), s=50, alpha=0.5)
plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=50)
plt.show()
```

Bài 2.2:

- Sinh viên làm tương tự bài 2.1

Bài 2.3:

- Sinh viên làm tương tự bài 2.1