Sử dụng các phương pháp học máy để phân loại tài liệu

Giảng viên hướng dẫn: TS. QUẢN THÀNH THƠ

Người thực hiện: Lê Kiến Trúc truc20mse23023@fsb.edu.vn

# Mô tả vấn đề:

* Cho một tập hợp các tài liệu với cấu trúc sau đây. Sử dụng các phương pháp đã học để phân cụm.

|  |  |
| --- | --- |
| **No.** | **Contents** |
| 1 | AABAAABBBAACD |
| 2 | ACDAAABBBCCCDDD |
| 3 | CCDDDCDDDDDCCC |
| 4 | ABCDEABCDEABCDE |
| 5 | ABBBBBBBBBBB |
| 6 | DEADEADDDDDE |
| 7 | BCBBBBCDCBBEEEEE |
| 8 | ABCAAAABBE |
| 9 | ABBEEABCEE |
| 10 | ABCBCBCBCDDD |

* Với cấu trúc dữ liệu nêu trên có thể sử dụng các phương pháp unsupervised learning để xử lý gom cụm. Các phương pháp thường được sử dụng là HAC và K-Means.
* Mỗi tập dữ liệu bao gồm 5 thành phần ABCDE và số lượng tài liệu chỉ 10 vì vậy theo dự đoán sử dụng phương pháp K-Means sẽ ra cho kết quả tốt nhất với thời gian vừa phải.
* Ngoài sử dụng TF-IDF để tạo ra vector mô tả dữ liệu, báo cáo này sẽ sử dụng thêm mạng nơ-ron dùng để tạo ra vector so sánh.
* Mã nguồn của báo cáo được lưu ở tập tin LeKienTrucFinal.ipynb

# Quá trình chạy phân tích dữ liệu:

* Sau khi dùng TF-IDF để thu ra ma trận vector và tập từ vựng

(array([[0.8538756 , 0.48792891, 0.12198223, 0.1339484 , 0. ],

[0.51514109, 0.38635582, 0.51514109, 0.56567525, 0. ],

[0. , 0. , 0.56400248, 0.8257731 , 0. ],

[0.40840975, 0.40840975, 0.40840975, 0.44847381, 0.54632926],

[0.09053575, 0.99589321, 0. , 0. , 0. ],

[0.2247474 , 0. , 0. , 0.86378111, 0.45096651],

[0. , 0.6865987 , 0.29425659, 0.10770749, 0.65604449],

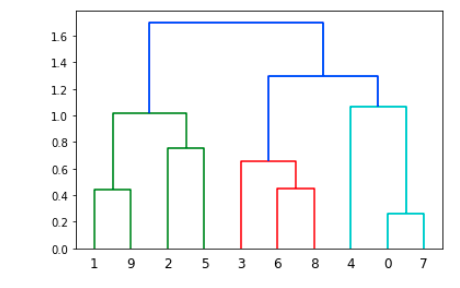
[0.8243439 , 0.49460634, 0.16486878, 0. , 0.22054478],

[0.30631423, 0.45947135, 0.15315712, 0. , 0.81951239],

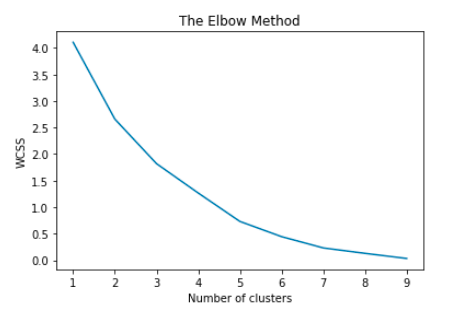
[0.15100923, 0.6040369 , 0.6040369 , 0.49746865, 0. ]]),

['a', 'b', 'c', 'd', 'e'])

* Để lựa chọn ra số lượng gom cụm tốt nhất. Thực nghiệm sử dụng Dendrogram để mô hình hóa dữ liệu.



* Theo kết quả thu được có thể gom cụm với số lượng 3 cụm.
* Thực nghiệm đồng thơi sử dụng K-Means và phương pháp Elbow để so sánh.



* Dự và kết trên có thể đưa ra kết luận số lượng Cluster tốt nhất là 3 ở khúc gập thứ 2.

# Kết quả:

* Kết quả sử dụng HAC của sklearn và thư viện K-Means ra kết quả như nhau, với silhouette score là 0.25

0.2539192732511972

Clustering data for 1

1 ACDAAABBBCCCDDD

2 CCDDDCDDDDDCCC

5 DEADEADDDDDE

9 ABCBCBCBCDDD

====================

Clustering data for 2

0 AABAAABBBAACD

4 ABBBBBBBBBBB

7 ABCAAAABBE

====================

Clustering data for 3

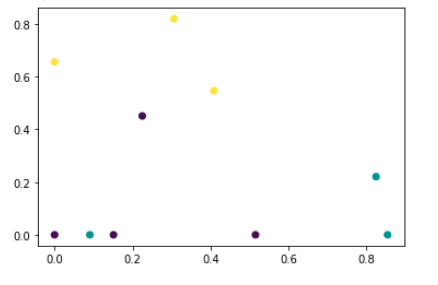
3 ABCDEABCDEABCDE

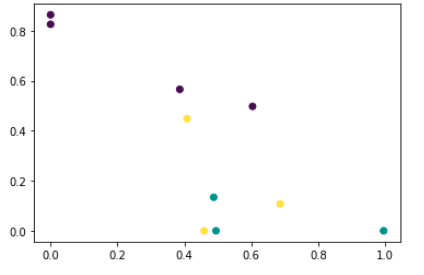
6 BCBBBBCDCBBEEEEE

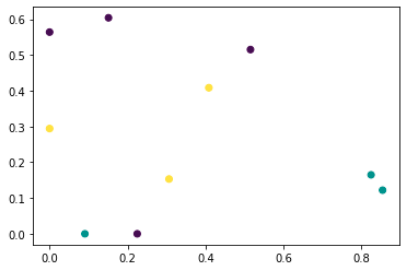
8 ABBEEABCEE

====================

Một vài biểu đồ mô tả không gian 2 chiều của dữ liệu:







# Cải tiến thực nghiệm:

* Qua quá trình nghiên cứu, tôi nhận thấy việc sử dụng các hàm thư viện HAC hay K-Means với các thông số mặc định có thể không đưa ra được kết quả tốt nhất về Cluster. Cho nên trong thực nghiệm này tôi có viết thêm 1 hàm tính toán sử dụng phương pháp vét cạn để cho kết quả với silhouette score lên đến 0.309

Clustering data for 1

0 AABAAABBBAACD

7 ABCAAAABBE

====================

Clustering data for 2

1 ACDAAABBBCCCDDD

2 CCDDDCDDDDDCCC

3 ABCDEABCDEABCDE

5 DEADEADDDDDE

9 ABCBCBCBCDDD

====================

Clustering data for 3

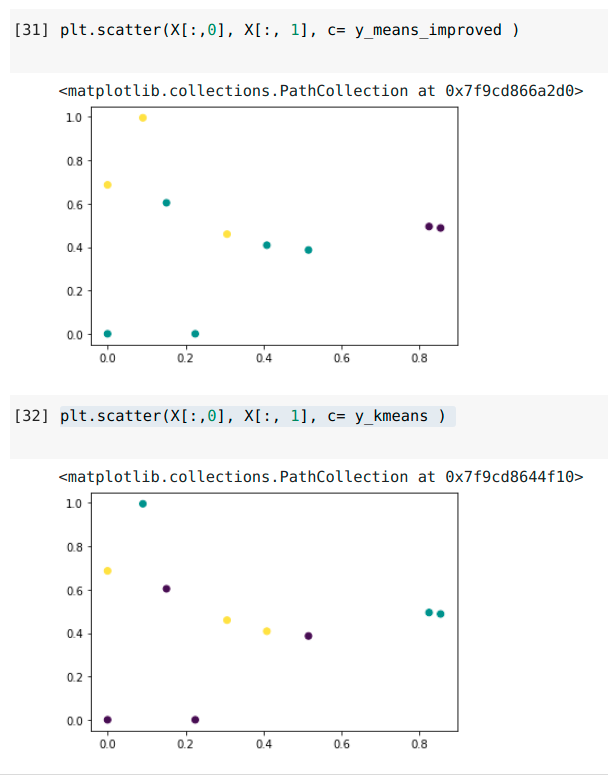
4 ABBBBBBBBBBB

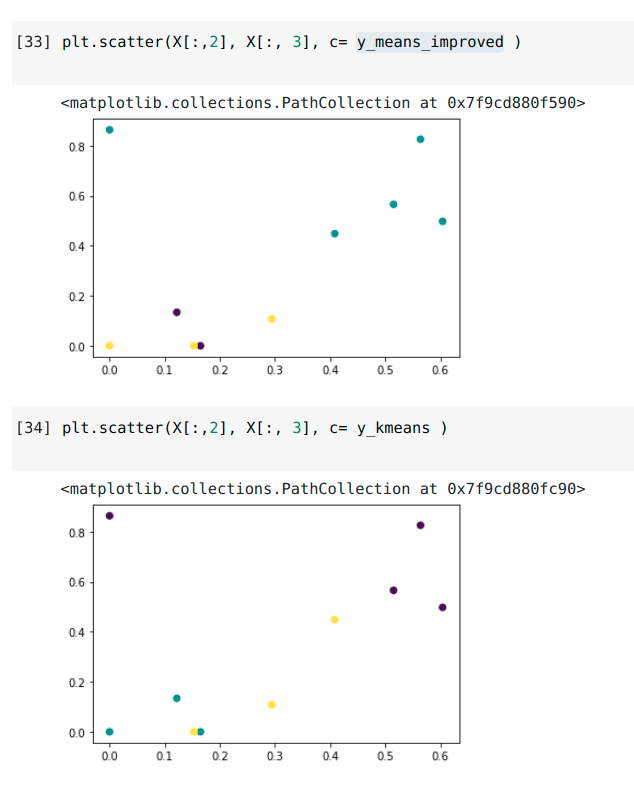
6 BCBBBBCDCBBEEEEE

8 ABBEEABCEE

====================

Khi so sánh kết quả K-Means vét cạn với thư viện cho kết quả hình ảnh phân cụm hợp lý hơn.





* Ngoài ra thực nghiệm còn sử dụng Word2Vec để xây dựng mô hình vector cho tài liệu thay vì sử dụng TF-IDF, và sử dụng thêm phương pháp K-Means để tính. Tuy nhiên kết quả thu được khác khá nhiều so với phương pháp TF-IDF

Clustering data for 1

1 ['A', 'C', 'D', 'A', 'A', 'A', 'B', 'B', 'B', 'C', 'C', 'C', 'D', 'D', 'D']

2 ['C', 'C', 'D', 'D', 'D', 'C', 'D', 'D', 'D', 'D', 'D', 'C', 'C', 'C']

3 ['A', 'B', 'C', 'D', 'E', 'A', 'B', 'C', 'D', 'E', 'A', 'B', 'C', 'D', 'E']

5 ['D', 'E', 'A', 'D', 'E', 'A', 'D', 'D', 'D', 'D', 'D', 'E']

6 ['B', 'C', 'B', 'B', 'B', 'B', 'C', 'D', 'C', 'B', 'B', 'E', 'E', 'E', 'E', 'E']

8 ['A', 'B', 'B', 'E', 'E', 'A', 'B', 'C', 'E', 'E']

9 ['A', 'B', 'C', 'B', 'C', 'B', 'C', 'B', 'C', 'D', 'D', 'D']

====================

Clustering data for 2

4 ['A', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B']

====================

Clustering data for 3

0 ['A', 'A', 'B', 'A', 'A', 'A', 'B', 'B', 'B', 'A', 'A', 'C', 'D']

7 ['A', 'B', 'C', 'A', 'A', 'A', 'A', 'B', 'B', 'E']

====================

# Kết luận:

* Với so sánh về phân cụm tài liệu với vector TF-IDF, kết quả khi sử dụng phương pháp HAC cho ra độ đo khá chính xác, tuy nhiên không phải là tuyệt đối cao nhất, có ưu điểm chạy nhanh.
* Thư viện K-Means nếu không sử dụng các biến để tăng số lượng lần lặp và số n\_init thì kết quả cũng tương tự HAC với cùng điểm số s-score. Tuy nhiên nếu áp dụng vào số lượng tài liệu lớn có thể dẫn đến việc thực hiện mất rất nhiều thời gian.
* Nếu sử dụng phương pháp vét cạn với K-Means cho ra số s-score tốt nhất có thể dùng phân cụm.
* Nếu sử dụng Word2Vec thay thế TF-IDF có thể cho ra kết quả khác, tuy không thể so sánh về điểm số. Nhưng kết quả phân cụm cũng khá tương đồng (0,7) và điểm (4) được sắp xếp thành cụm riêng với toàn bộ các cụm còn lại. Tuy nhiên sự chênh lệch về số lượng tài liệu trong cụm đôi lúc không phải mục tiêu của việc phân cụm.
* Với các kết quả thực nghiệm trên, chúng ta có thể đưa ra kết luận việc sử dụng HAC và thư viện K-Means có những hiện thực rất tốt cho việc gom cụm tài liệu.