

[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)[ThucHanhTriTueNhanTao / Knn.md](#) 

ThenScu Thêm hướng dẫn chi tiết về thuật toán K-Means



27518a0 · now



108 lines (52 loc) · 4.8 KB

[Preview](#)[Code](#)[Blame](#)[Raw](#)

1. Ý tưởng cốt lõi của K-Means

Mục tiêu của K-Means là chia một tập dữ liệu lớn thành k nhóm (cụm) riêng biệt, sao cho các điểm trong cùng một nhóm thì giống nhau (gần nhau về khoảng cách) nhất có thể.

Thuật toán hoạt động theo kiểu "lặp đi lặp lại" (iterative) qua 2 hành động chính: Chọn phe (Gán nhãn) và Dời đô (Cập nhật tâm).

2. Giải thích chi tiết theo Code

Giai đoạn chuẩn bị (Bước 1 - 3)

Phần này tạo ra "đề bài". Bạn tạo ra 500 điểm dữ liệu giả lập tập trung quanh 3 vị trí $(2,2)$, $(9,2)$ và $(4,9)$.

Mục đích: Để kiểm tra xem lát nữa thuật toán có tự tìm lại được 3 vị trí trung tâm này không.

Giai đoạn khởi tạo (Bước 4: kmeans_init_centers)

Code: `np.random.choice(X.shape[0], n_cluster, replace=False)`

Ý nghĩa: Khi bắt đầu, máy tính không biết tâm cụm nằm ở đâu. Nó nhăm nntp chọn bừa 3 điểm bất kỳ trong dữ liệu làm "Tâm cụm tạm thời".

Giai đoạn "Chọn phe" (Bước 5: kmeans_predict_labels)

Đây là bước quan trọng nhất. Mỗi điểm dữ liệu sẽ tính toán xem mình gần tâm nào nhất.

Code:

$D = \text{cdist}(X, centers)$: Tính khoảng cách từ mỗi điểm đến cả 3 tâm.

$\text{np.argmin}(D, axis=1)$: Tìm xem khoảng cách nào ngắn nhất.

Ví dụ: Điểm A cách Tâm 1 (5m), Tâm 2 (10m), Tâm 3 (20m). → A chọn về phe Tâm 1 (nhãn 0).

Kết quả: Không gian được chia thành các vùng lãnh thổ (Voronoi regions).

Giai đoạn "Dời đô" (Bước 6: kmeans_update_centers)

Sau khi các điểm đã chọn phe xong, các tâm cụm nhận ra vị trí hiện tại của mình chưa chuẩn (vì lúc đầu chọn bừa). Tâm cụm cần di chuyển vào chính giữa đám đông ủng hộ mình.

Code: $\text{np.mean}(Xk, axis=0)$

Toán học: Tính trung bình cộng tọa độ (x, y) của tất cả các điểm thuộc phe k . Kết quả trung bình này chính là vị trí mới của tâm cụm.

Giai đoạn kiểm tra (Bước 7: kmeans_has_converged)

Ý nghĩa: Kiểm tra xem vị trí tâm mới và tâm cũ có giống hệt nhau không.

Nếu giống (Hội tụ): Nghĩa là các tâm đã nằm im, không di chuyển nữa → Thuật toán dừng.

Nếu khác: Nghĩa là vẫn còn sự xáo trộn → Tiếp tục lặp.

3. Quy trình chạy thực tế (Bước 9: kmeans)

Hàm kmeans là "nhạc trưởng" điều phối toàn bộ quá trình trong vòng lặp while True:

Lặp lần 1:

Từ 3 tâm chọn bừa → Các điểm chia phe (Labels).

Từ phe mới chia → Tính lại vị trí tâm (Centers) tốt hơn.

Visualize: Bạn sẽ thấy các tâm di chuyển một đoạn khá xa về phía các cụm dữ liệu.

Lặp lần 2, 3, 4...:

Lại chia phe lại dựa trên tâm mới (một số điểm ở rìa sẽ đổi phe).

Lại tính tâm mới.

Visualize: Các tâm nhích dần từng chút một vào trọng tâm của đám mây dữ liệu.

Dừng (Break):

Khi tính ra tâm mới trùng khớp hoàn toàn với tâm cũ.

Tóm tắt trực quan

Hãy tưởng tượng bạn mở 3 quán cafe.

Bước 4: Bạn cắm đại 3 cái cờ ở 3 vị trí ngẫu nhiên.

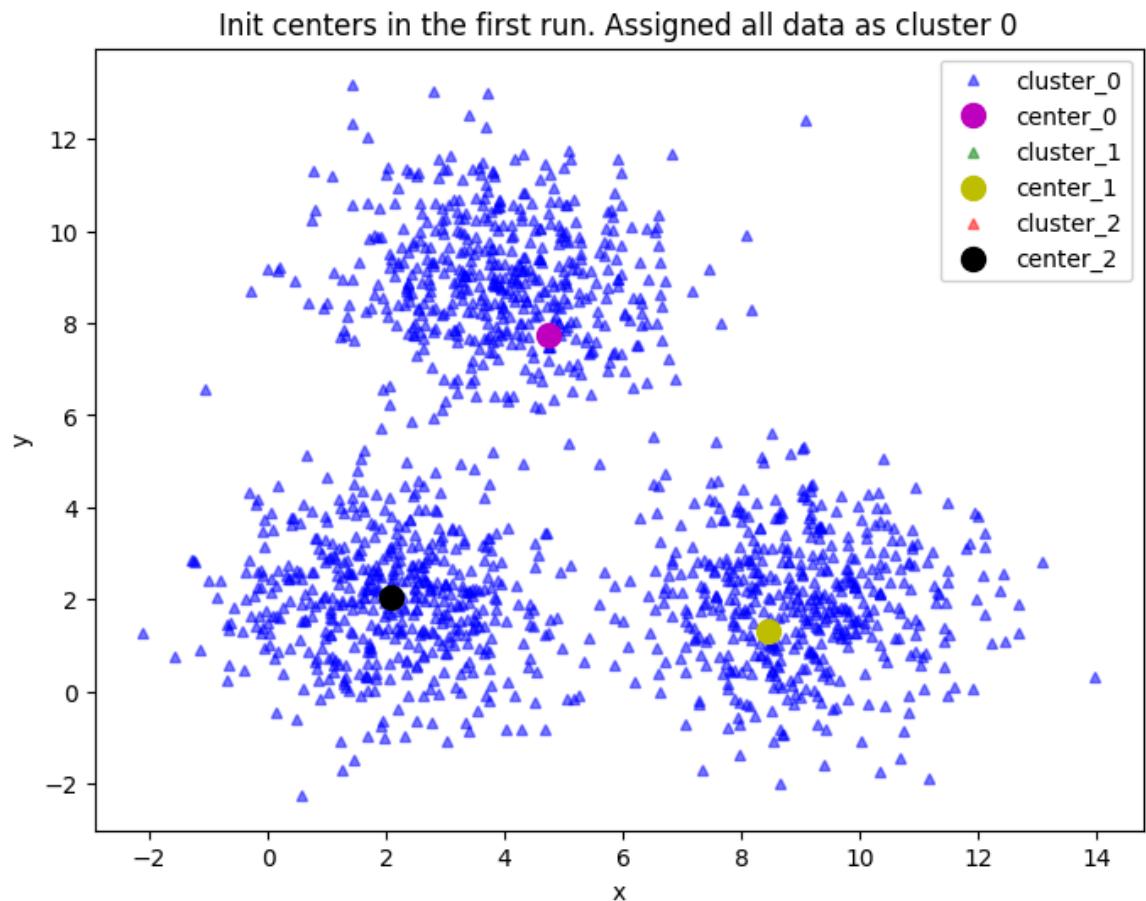
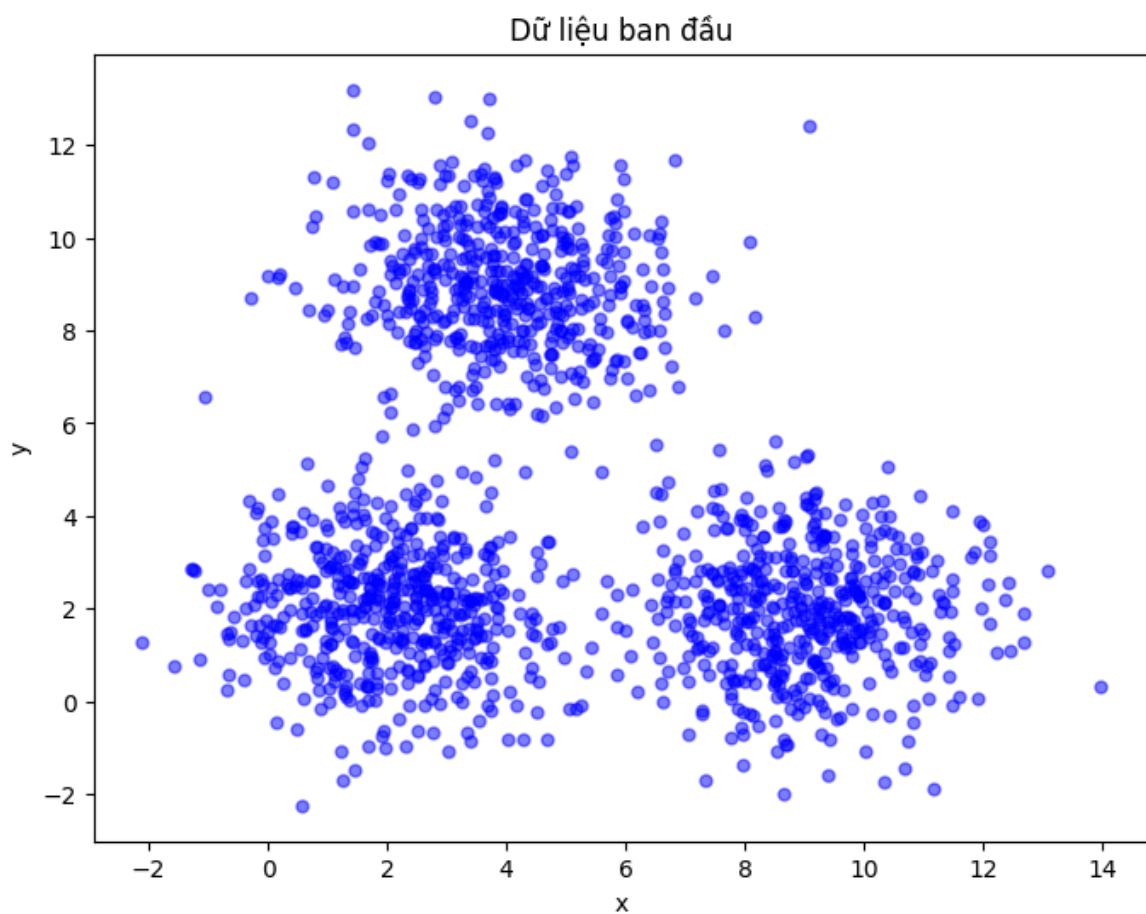
Bước 5: Khách hàng sẽ đi đến quán nào gần họ nhất.

Bước 6: Bạn thấy quán 1 đông khách nhưng khách toàn đi từ phía Bắc xuống. Bạn dời quán 1 lên phía Bắc để nằm giữa khu dân cư đó.

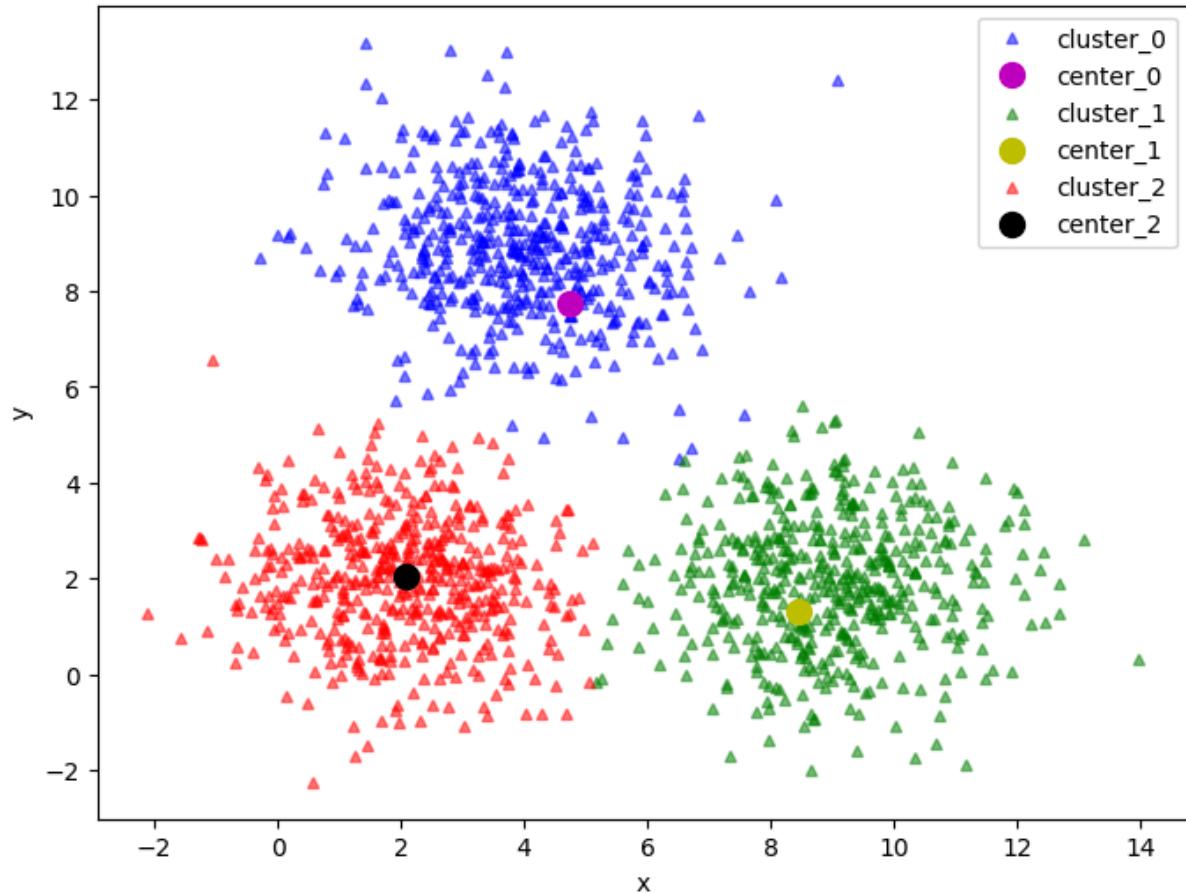
Lặp lại cho đến khi quán nằm đúng vị trí tối ưu nhất (trung tâm khu dân cư).

Markdown## 😊 Giải thích ý nghĩa kết quả (Output Analysis)

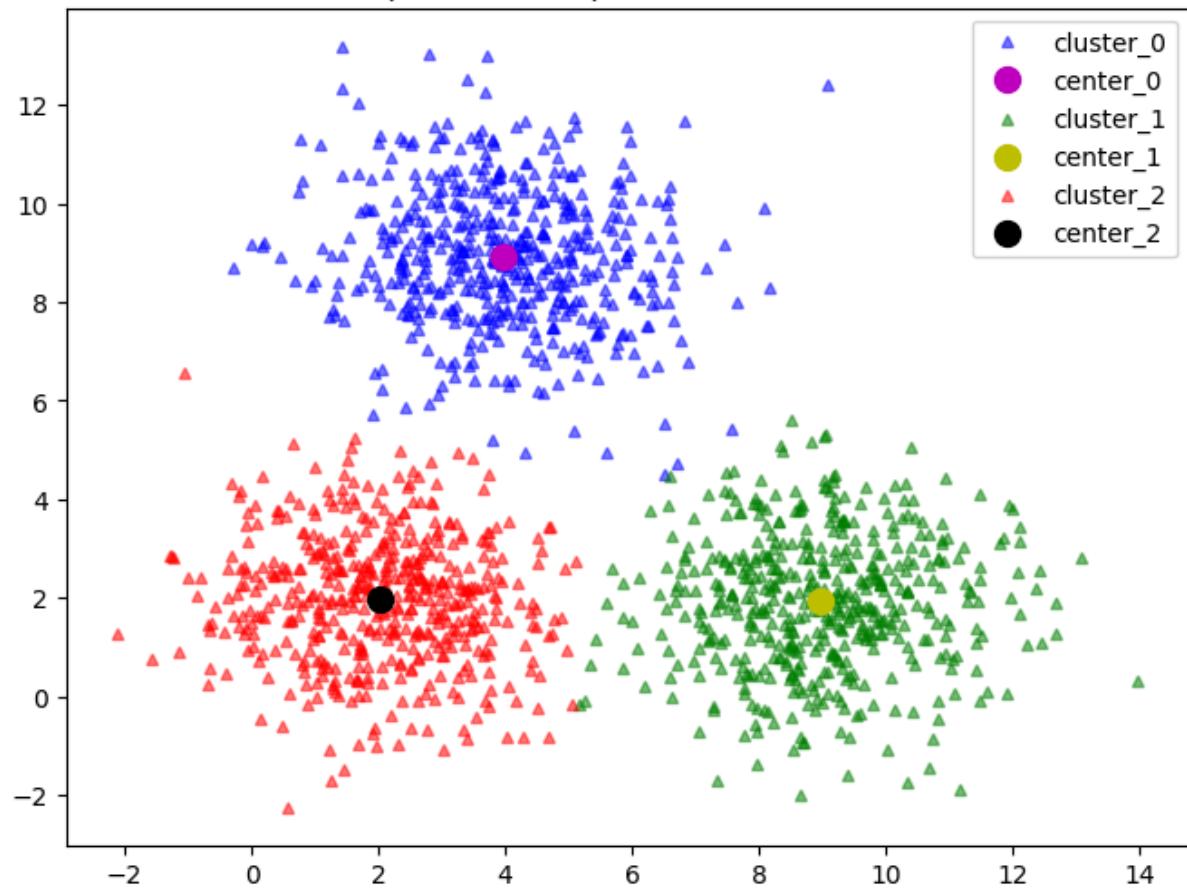
Khi chạy chương trình, bạn sẽ nhận được các dòng thông báo kết quả. Dưới đây là giải thích chi tiết ý nghĩa của từng con số:

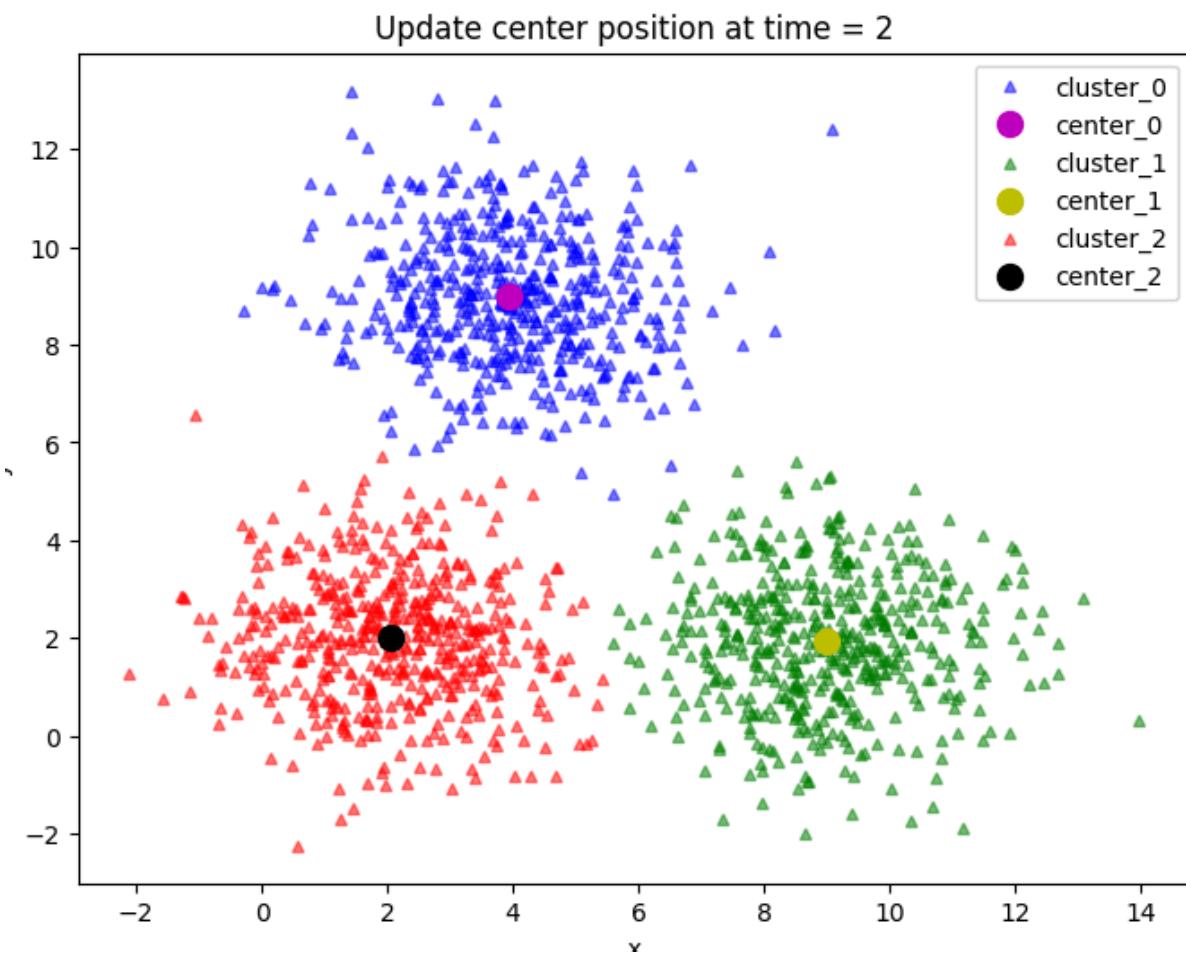
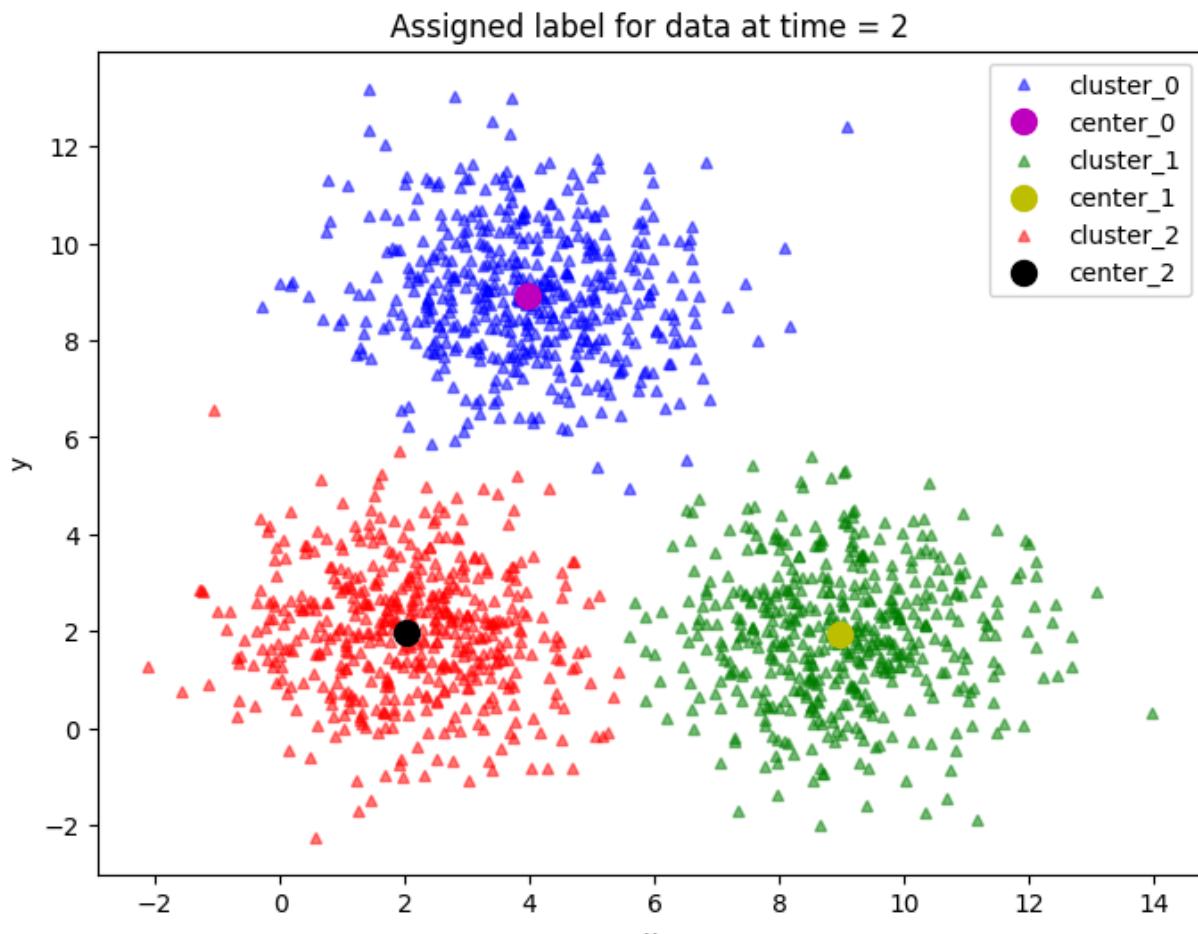


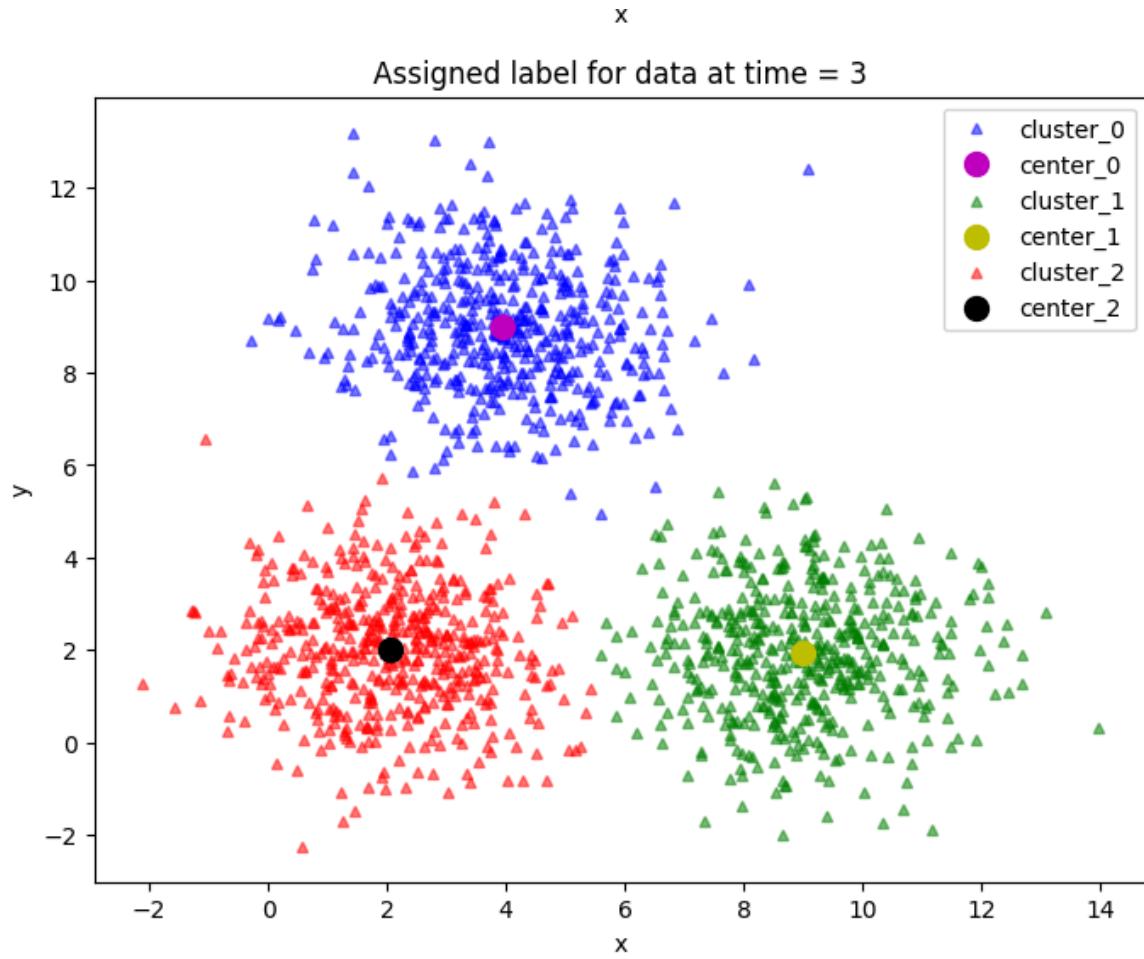
Assigned label for data at time = 1



Update center position at time = 1







Done! Kmeans has converged after 2 times

Final centers:

```
[[3.94416198 8.98396863]
 [8.98259068 1.95213177]
 [2.05198992 1.99612428]]
```