حامد کمترزاده ۹۸۱۲۷۶۲۴۱۸

- هر image ابعاد 70x80 دارد ← هر sample 5600 تا بعد دارد (از مقادیر پیکسل های تصویر به عنوان ویژگی ها استفاده می‌کنیم) (تصاویر gray scale اند)

Kmeans:

- از کتابخانه sklearn استفاده می‌کنیم.
- امتیاز Rand Index = 0.97 (انتخاب seed مناسب، k=41 دقیقاً برابر با ۴۱)

Agglomerative:

Average link:

- از کتابخانه sklearn استفاده می‌کنیم.
- امتیاز rand_index = 0.95 (k=41)

Single link:

- امتیاز rand-index = 0.63 (k=60) ← over clustering

(با انتخاب k=41 خوشه بندی نامطلوب برای over clustering قابل تحمل است)

Complete link:

- امتیاز rand-index = 0.97 (k=41)

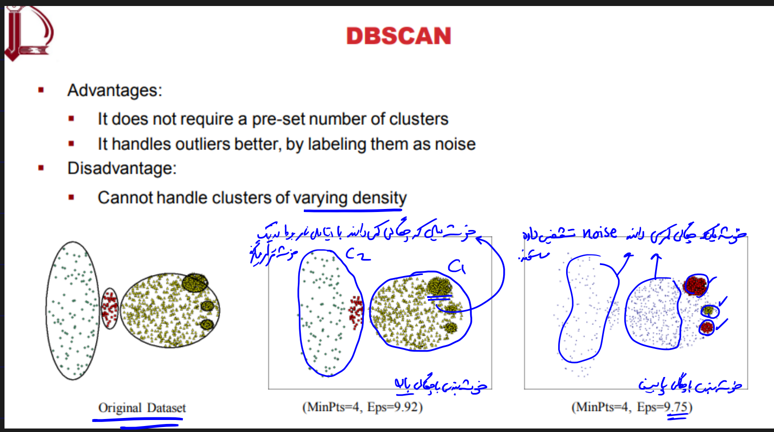
DBSCAN:

- از کتابخانه sklearn استفاده می‌کنیم. (از فاصله اقلیدسی استفاده می‌کنیم)
- برای انتخاب epsilon از الگوریتم k-nearest neighbors استفاده می‌کنیم. (توضیحات استفاده از KNN در لینکی که در قسمت DBSCAN در Notebook قرار داده آمده است.)
- بهتر است برای DBSCAN داده ها را نرمال کنیم (تا انتخاب ε راحت‌تر شود)
- امتیاز: rand index = 0.82 (min points = 4, epsilon = 55)

← داده‌های زیادی noise تشخیص داده می‌شوند. راه حل این مشکل که ناشی از متفاوت بودن چگالی خوشه‌ها است در قسمت بعد توضیح داده شده است.

بهبود DBSCAN:

- از مشکلات اصلی الگوریتم dbscan این است که خوشه‌بندی با چگالی‌های متفاوت را پشتیبانی نمی‌کند.



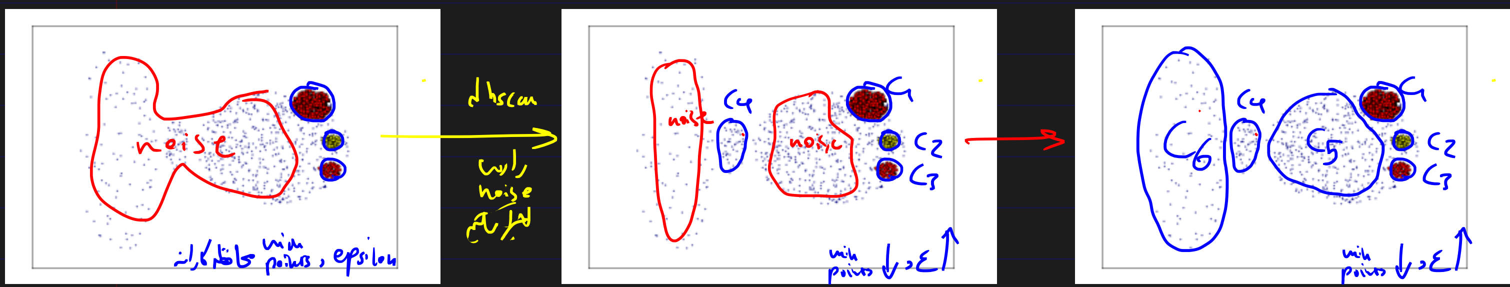
یک راه حل برای رفع این مشکل این است که الگوریتم DBSCAN را برای چند epsilon و min points مختلف اجرا کنیم.

- این مشکل در دیتاست مسئله ما هم دیده می‌شود | تعداد زیادی داده با لیبل -1 به عنوان noise شناخته می‌شوند)

- برای رفع این مشکل در دیتاست داده شده از این روش استفاده می‌کنیم که به صورت محافظه‌کارانه epsilon و min points را انتخاب می‌کنیم (ε کم و min points زیاد)

و در مرحله بعد روی دیتاهایی که در مرحله قبل noise تشخیص داده شده دوباره DBSCAN می‌زنیم اما epsilon را افزایش می‌دهیم (و احتمالاً min points را هم کاهش دهیم)

با اینکار خوشه‌هایی با چگالی متفاوت بدست می‌آیند. (اگر خوشه‌ها کنار هم قرار نگیرند).



- این روش در notebook Enhanced DBSCAN پیاده‌سازی شده

به این صورت که در هر مرحله ابتدا DBSCAN اعمال و بر اساس نمودار KNN برای انتخاب epsilon رسم شده و ε و min points انتخاب شده و خوشه‌بندی انجام می‌شود و مجدداً نمودار KNN برای دیتاهایی که در مرحله قبل noise تشخیص داده شده‌اند رسم می‌شود و ε و min points جدید برای این noise ها انتخاب شده و خوشه‌بندی روی noise ها انجام شده و مراحل قبل برای noise های جدید تکرار می‌شود.

با انتخاب ε = 55 و min points = 4 برای مرحله اول و ε = 60 و min points = 3 در مرحله دوم امتیاز rand index به 0.97 می‌رسد و هم ترازنه noise کمتری تشخیص داده شد.

پیاده‌سازی Rand Index:

$TP+FP$ = تعداد کل جفت‌هایی که در یک cluster قرار دارند

(same cluster)

$$= \sum_{i=1}^{k} \binom{N_i}{2}$$

N_i = population in cluster i,
k = number of clusters

TP = similar Idams in a cluster (Count)

$$FP = (TP+FP) - TP$$

$$FN = FN_{i_1} + FN_{i_2} + \dots + FN_{i_t}$$

$$= \left(\binom{N_1}{2} - TP_1\right) + \left(\binom{N_2}{2} - TP_2\right) + \dots + \left(\binom{N_t}{2} - TP_t\right)$$

false negative مربوط به اسم 2 در 1 = تعداد کل جفت اشیاء از نوع 1 (similar) − true positive اسم 1

false negative: similier in diffe rent clusters

true positive: Similar in same cluster

$$TN = N - (FN + TP + FP)$$