به نام خدا

گزارش فاز اول

پروژه هوش محاسباتی

پوریا ناظمی و طهورا سعیدی

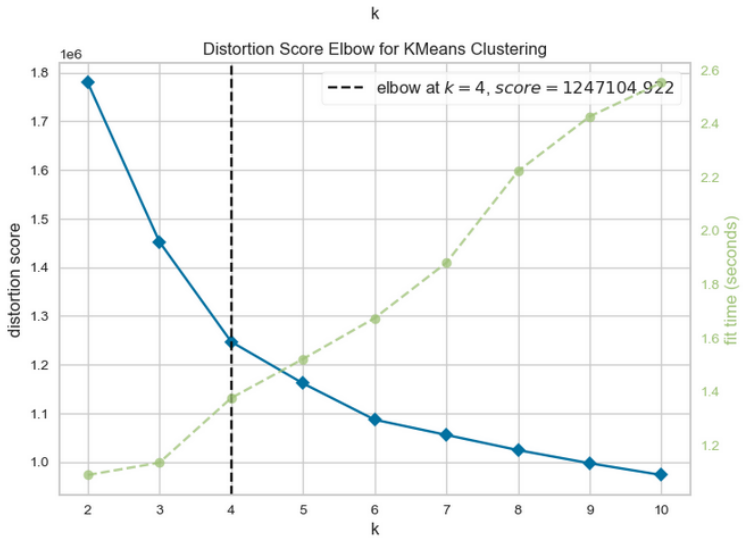
در این مسئله ما با داده هایی از تصاویر که لیبل شئ داخل تصویر مشخص است، روبرو هستیم. تصاویر موجود هدف خوشه بندی داده ها بر حسب دامین تصاویر است. منظور از دامین نوع تصویرموجود از آن شئ است. که برای مثال ممکن است بخشی از تصاویر نقاشی، بخشی عکس دوربین وغیره باشند. خوشه بندی به صورت unsupervised است و از لیبل دامین داده ها اطلاعی نداریم.

چالش هایی که با آنها روبرو هستیم، خوشه بندی به شکل مناسب است و پس از آن پیدا کردن تعداد دامین ها و مپ کردن خوشه ها با لیبل های دامین تصاویر است.

به منظور خوشه بندی و ارزیابی داده ها، راهکارها و الگوریتم های مختلفی را امتحان کرده ایم. در برخی از الگوریتم ها همانند kmeans نیازمند دانستن تعداد مناسب خوشه ها هستیم. به این منظور راهکارهای مختلفی را امتحان میکنیم.

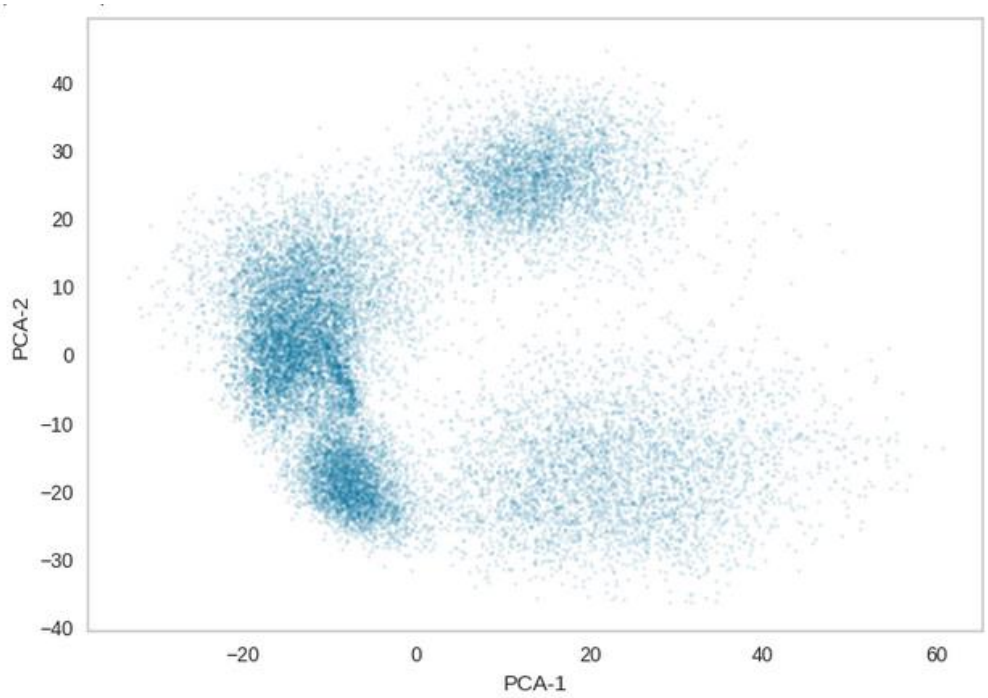
در یکی از شیوه های مورد استفاده قرار گرفته به این شکل پیش رفتیم که داده ها را بر حسب لیبل نوع شئ تصویر جدا کردیم و روی هر بخش جداگانه الگوریتم kmeans را با k های مختلف اجرا کردیم. و با WCSS ارور مینیمایز را تشخیص میدهیم که در کدام k صورت گرفته است. چرا که به این شکل تعداد دامین ها را در داده های محدود بهتر میتوان مشخص کرد.

نتایج نشان میدهد که در نیمی از اشیا k برابر با ٤ و در نیمی دیگر k برابر با ٥ ، مطلوب بوده است. این مورد نشان میدهد که میتوان تخمینی از ٥ دامین داشت، اما تنها به همین یک معیار نمیتوان اکتفا کرد. نکته دیگری که وجود دارد این است که گرچه در داده های تک لیبل با k برابر با ٤ یا ٥ نتیجه بهتری گرفتیم، اما نمیتوان تضمین کرد که با این تعداد دسته و همان الگوریتم kmeans بر روی کل داده ها نیز لزوما نتیجه مطلوب خواهد داد چرا که در مجموع داده ها ممکن است نتایج متفاوت حاضر شود.



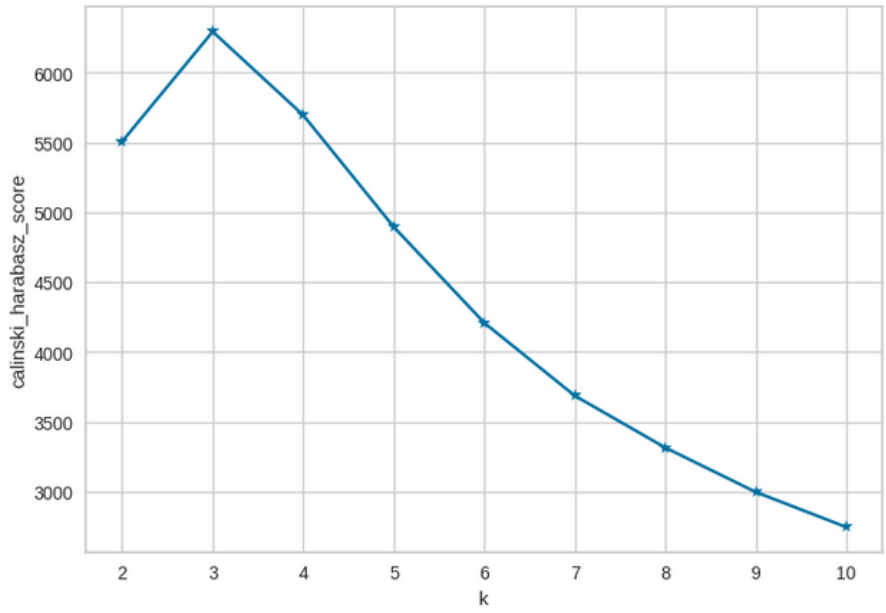
نمونه ای از تست با k های مختلف و گزینش k برای هر image label

وقتی با PCA داده ها رو در دو بعد نمایش میدهیم، مشخص میشود که به طور کلی داده ها را میتوان به ۳ دسته تقسیم کرد، اما نکته ای که وجود دارد این است که تراکم و چگالی داده ها در دسته ها متفاوت است. به طوریکه این ۳ دسته کلی را میتوان به زیر بخش هایی تقسیم کرده که هر یک کلاستر جداگانه ای باشند.

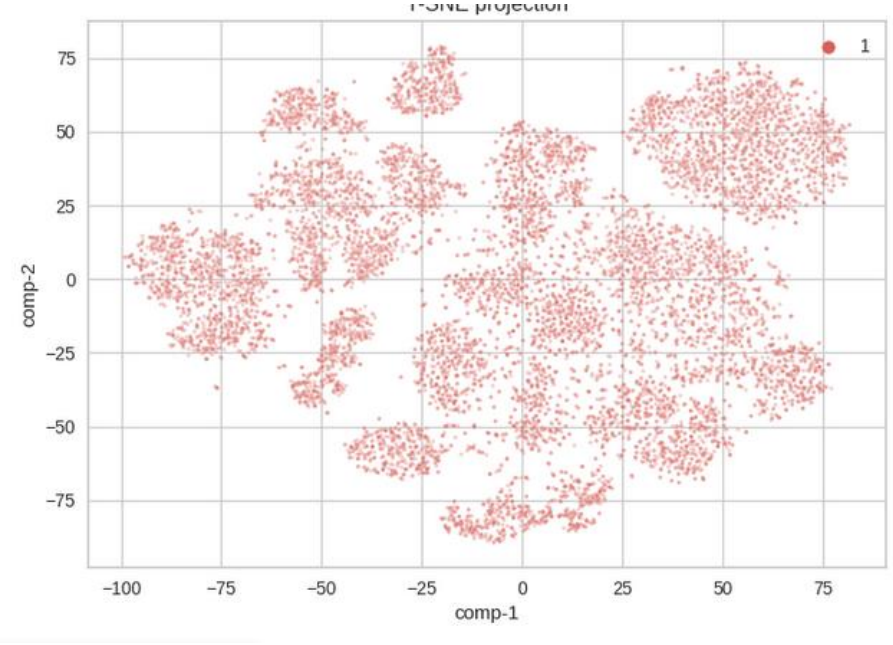


با توضیحات گفته شده درباره نوع توزیع داده ها، اگر ما با توجه به این شکل کلی داده ها، بر روی همه داده ها اگر بخواهیم برای اجرای الگوریتم kmeans، k مناسب را بیابیم، با متریک های معمول مثل calinski، k برابر با ۳

بیشترین امتیاز را میگیرد، اما نکته ای که وجود دارد اینگونه متریک معیار تشخیص امتیاز و ارزیابی خوشه بندی، حداکثر کردن فاصله بین کلاستری و مینمم کردن فاصله درون کلاستری است. این نوع ارزیابی در خصوص داده های ما، با نحوه قرارگیری نشان داده شد در PCA آنها، طبیعتا مقدار ۳ کلاستر را بهتر ارزیابی میکند، چرا که در این حالت مراکز کلاستر فاصله بیشتری دارند و داده های دور مراکز متراکم ترند. این مورد بی توجه به این است که درون یک کلاستر ممکن است بخشی از داده ها که جنس و چگالی متفاوتی دارند، قرار گرفته باشند. پس این نوع ارزیابی و متریک نمیتواند برای تشخیص تعداد دامین ها و خوشه بندی به تعداد بدست آمده مناسب باشد.
در واقع وقتی الگوریتم kmeans را با k برابر با ۳ روی کل داده ها اجرا کنیم، یک کلاستر شامل تعدادی زیاد از داده ها می شود، (۱۲ هزار داده) که اگر نمودار TSNE این تک کلاستر را پلات کنیم، پراکندگی داده ها مشهود است.

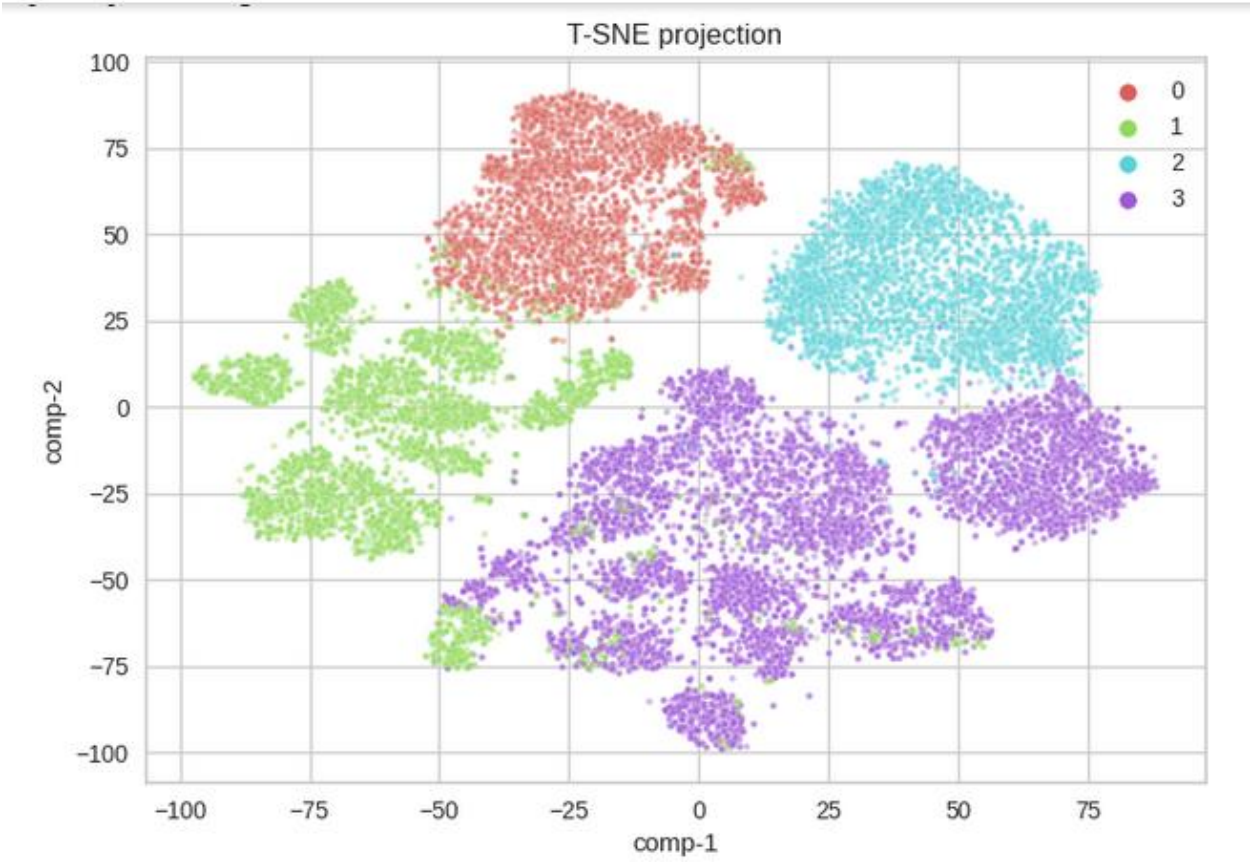


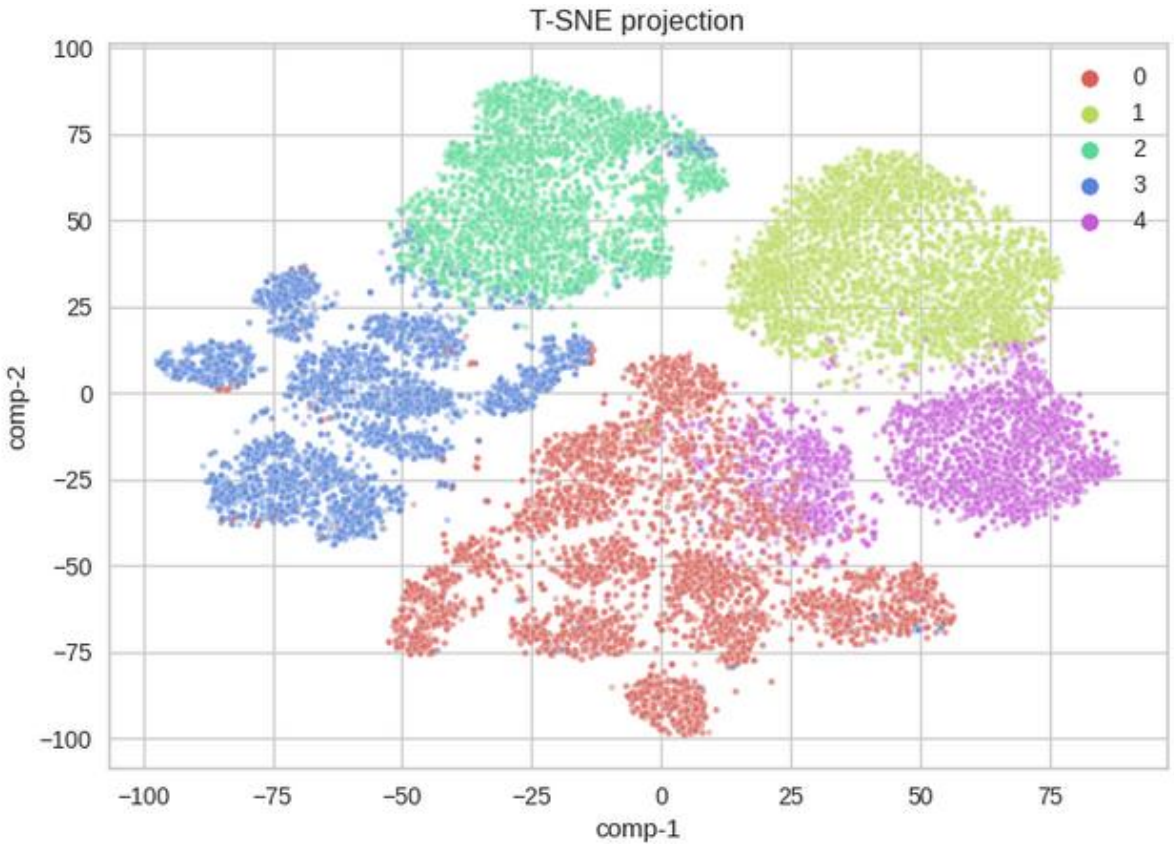
پیدا کردن k مناسب روی کل داده ها

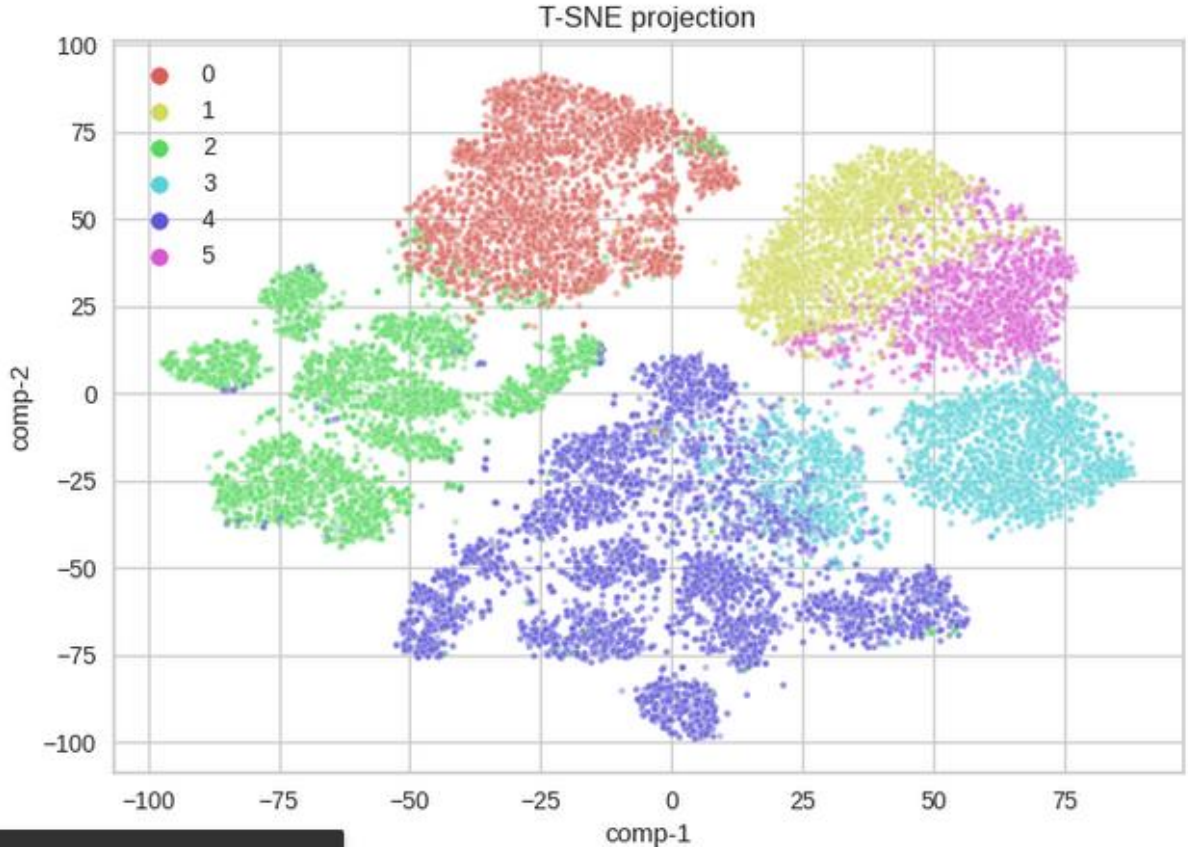


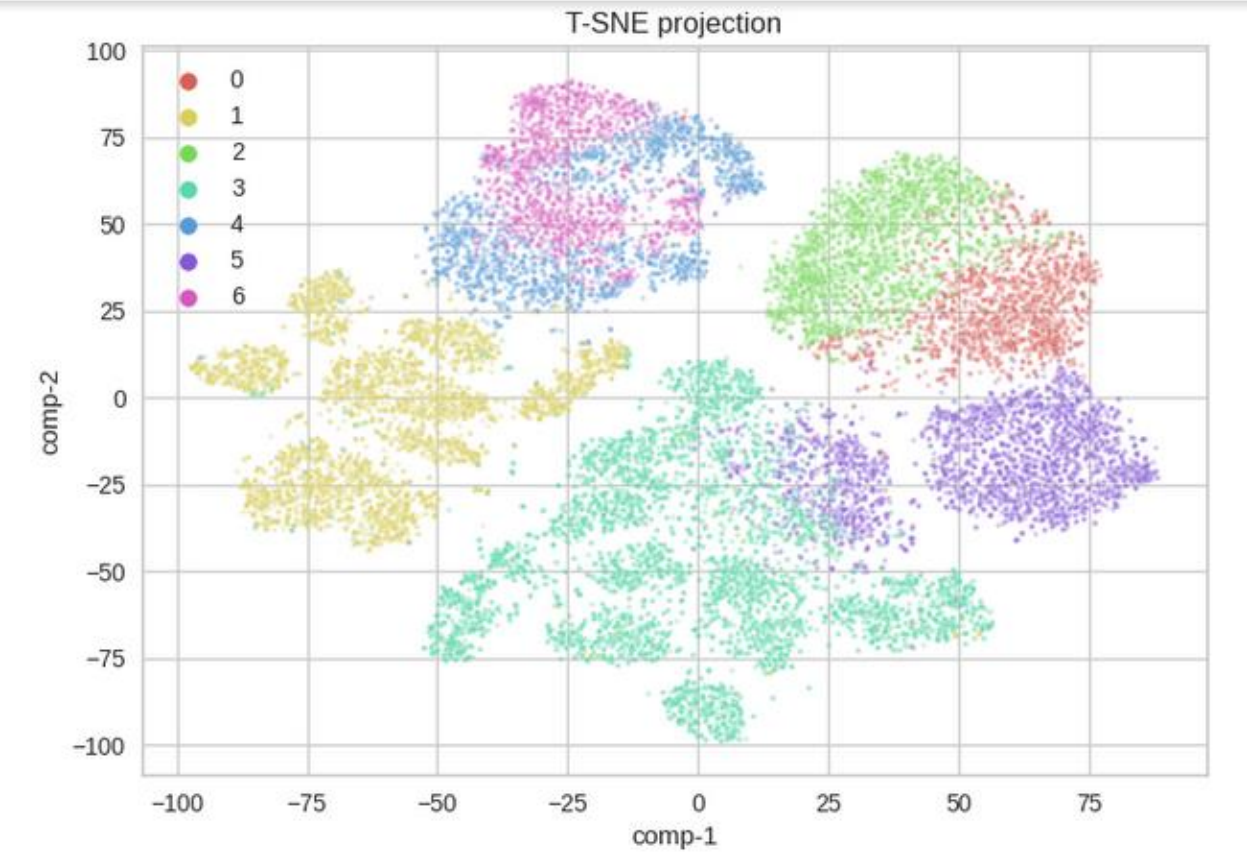
پراکندگی داده ها در بزرگترین کلاستر پس از کلاستربندی با k برابر با ۳

با توضیحات ذکر شده الگوریتم kmeans را برای بازه ای از k ها (۷-٤) که با موارد گفته شده معقول تر است، بر روی کل داده ها اجرا میکنیم و با TSNE پلات میکنیم و کیفیت خوشه بندی را ارزیابی میکنیم.



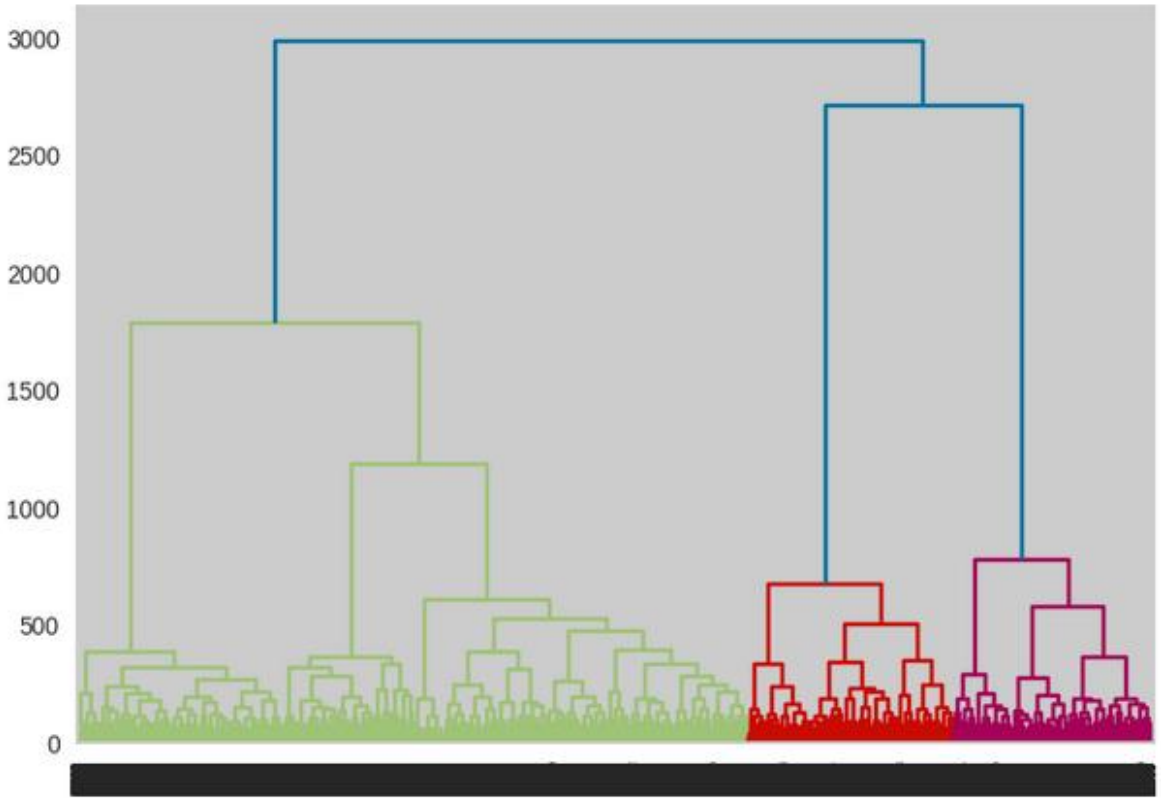

T-SNE projection
0
1
2
3
4
comp-2
comp-1
100
75
50
25
0
−25
−50
−75
−100
−100
−75
−50
−25
0
25
50
75


T-SNE projection
0
1
2
3
4
5
comp-2
comp-1
100
75
50
25
0
−25
−50
−75
−100
−100
−75
−50
−25
0
25
50
75



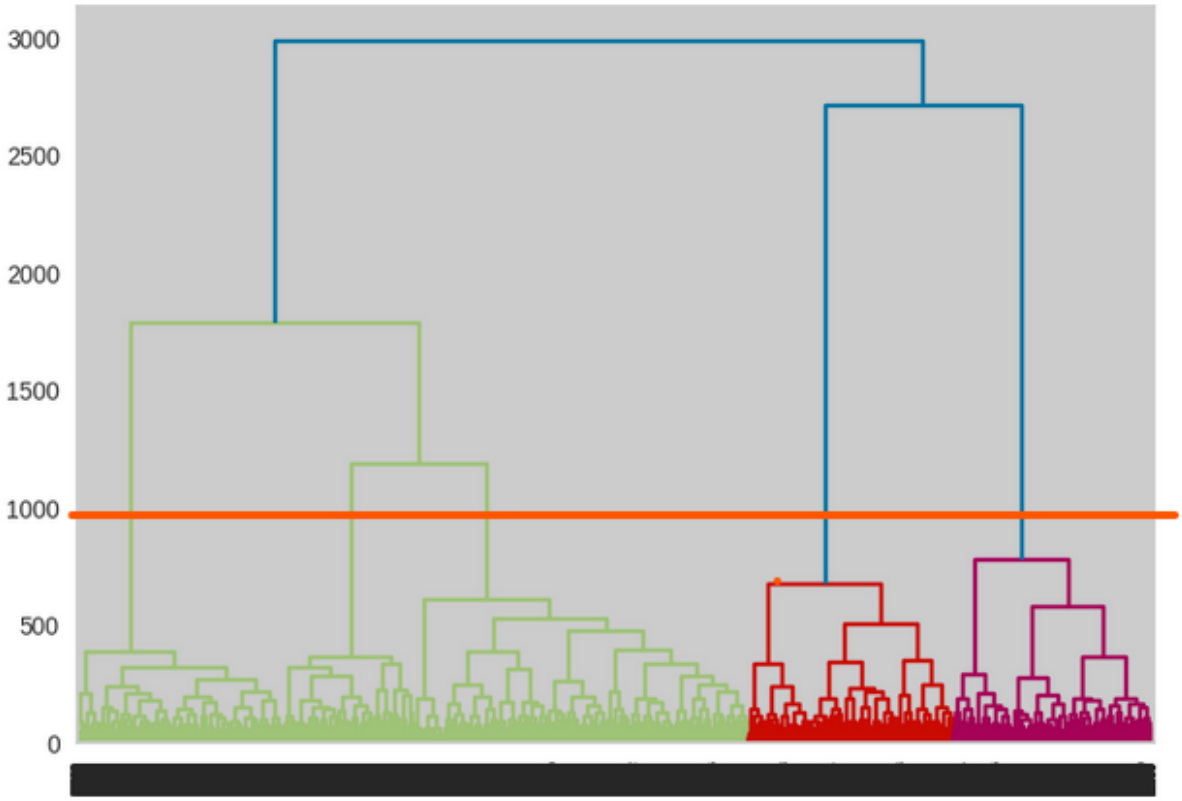
نکته ای که وجود دارد این است که TSNE توزیع داده ها را تقریبا در پنج خوشه نشان میدهد. آنچه که مشخص می شود این است که بایستی در نهایت الگوریتم و پارامتر هایی را پذیرفت که بهتر داده ها را در این ۵ دسته تقسیم کنند و اگر الگوریتم در یک مرحله به این مهم نرسید، میتوان راهکارهایی همچون کلاستر بندی مجدد کلاسترها و یا ادغام کلاسترها در نظر گرفت.

از سوی دیگر یکی دیگر از الگوریتم های خوشه بندی الگوریتم های سلسه مراتبی است. یکی از نمودار های مورد استفاده در بحث الگوریتم های خوشه بندی سلسله مراتبی، نمودار دندوگرام است. در این رویکرد، که یک رویکرد از پایین به بالاست، ابتدا هر دیتاپوینت به تنهایی یک کلاستر در نظر گرفته میشود، سپس هر کلاستر با کلاسترهای دیگر مقایسه میشود و بر مبنای شباهت کلاستر ها ادغام میشوند، این اقدام تا تبدیل کل داده ها به یک کلاستر ادامه میابد. هر چه طول خط های عمودی قبل از ادغام بیشتر باشد، نشان دهنده این است که آن دو کلاستر شباهت کمتری به هم داشته اند.



نمودار دندوگرام

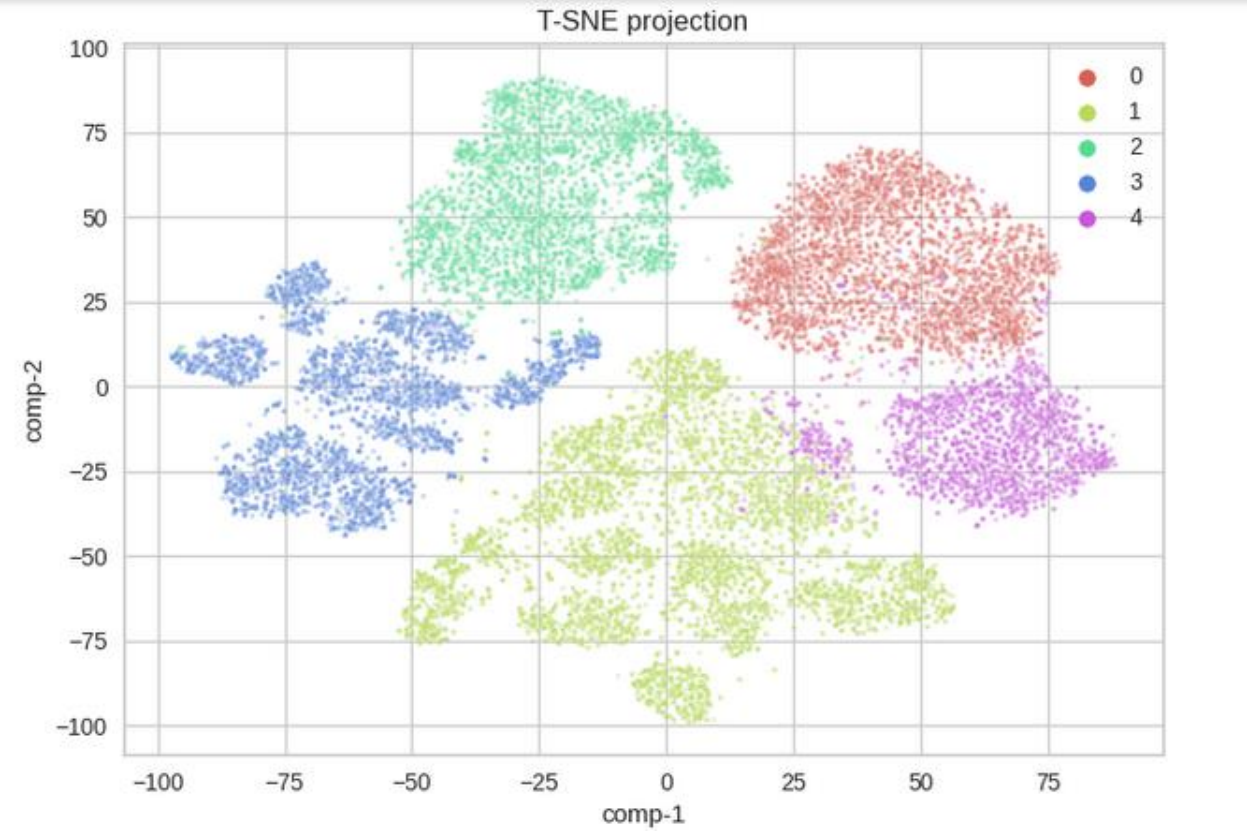
وقتی این نمودار را برای داده های مسئله خود استفاده کردیم، نتیجه با PCA داده ها و اجرای kmeans با k برابر با ۳ همخوانی دارد. در واقع وقتی این نمودار را در ۱۰۰۰ برش میزنیم، داده ها در ابتدا ۲ دسته میشوند و یک دسته به ۲ کلاستر تقسیم میشود، اما دسته دیگر در ابتدا به ۲ دسته و سپس یکی از دسته ها به ۲ دسته دیگر تقسیم میشود. در نهایت ۵ خوشه در این عمق تشکیل میشود که روند تقسیم بندی آنها با نحوه توزیع بررسی شده از قبل همخوانی دارد. (۳ دسته کلی که یکی از این ۳ دسته دارای زیر بخش است که خود میتوانند کلاسترهای جداگانه ای شوند.)



دوندوگرام کات شده

بنابراین الگوریتم های سلسله مراتبی نیز میتوان برای این نوع از داده ها روش خوشه بندی مناسبی باشند. در ابتدا با چند پارامتر مختلف این نوع الگوریتم ها را تست میکنیم و سپس، به صورت ترکیبی نیز الگوریتم های kmeans و سلسله مراتبی را استفاده خواهیم کرد و عملکرد هر یک را می سنجیم.

ابتدا الگوریتم را با agglomerative clustering و با تعداد ۵ کلاستر، کلاستر میکنیم. نمودار TSNE این خوشه بندی به شکل زیر می شود.

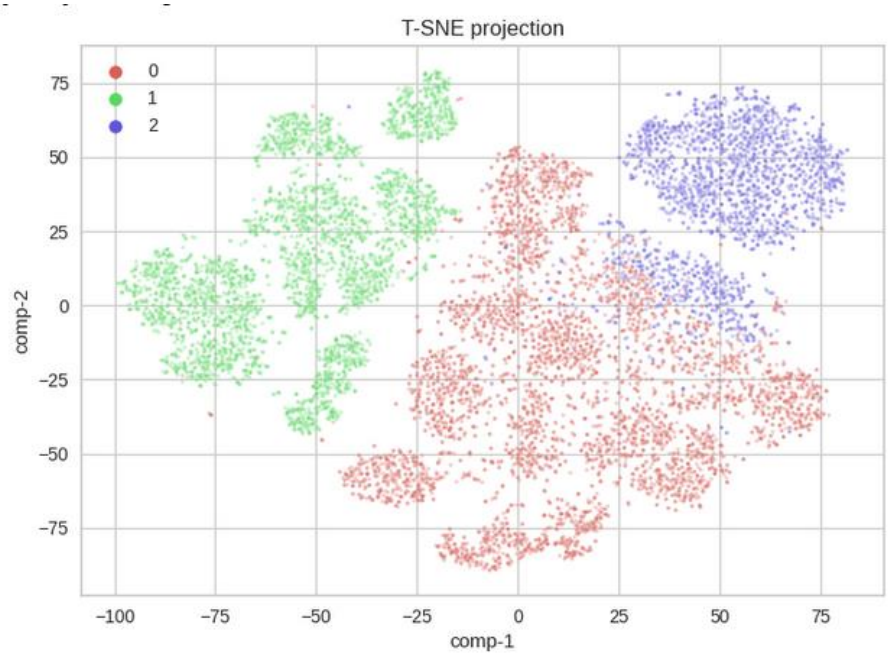


خروجی حاصله تا حدی بهتر از kmeans با k برابر با ۵ است. این الگوریتم را با متد های linkage مختلف و تعداد کلاستر ۶ نیز ران کردیم و بهترین نتیجه در این حالت فعلی حاصل شد. (الگوریتم انتخابی نهایی خوشه بندی)

روش ترکیبی دیگری که به کار گرفتیم ایجاد یک overclustring است و سپس ادغام و merge کلاستر ها. ابتدا kmeans با k برابر با ۷ بر روی داده ها اجرا کنیم، سپس با معیار از فاصله مرکز خوشه ها، با استفاده agglomerative کلاسترینگ، دوباره کلاستری بندی میکنیم.

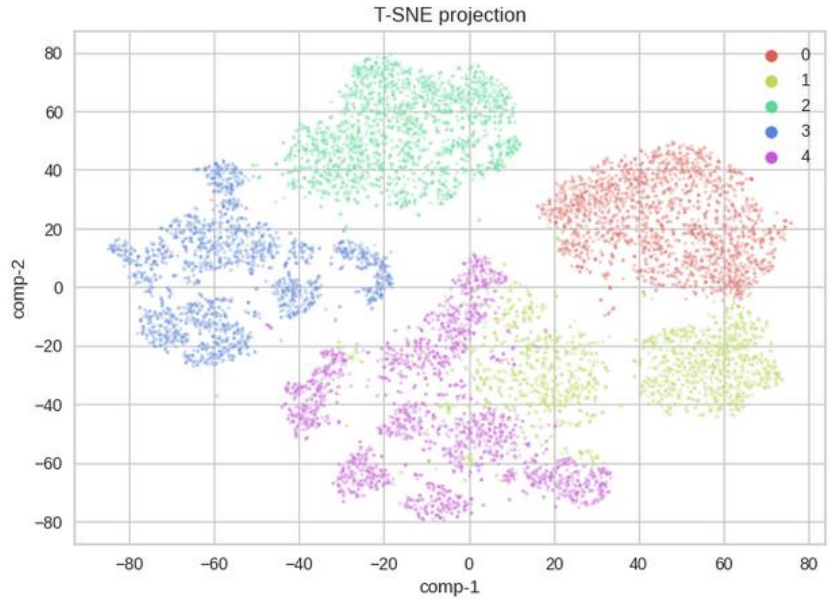
نتیجه پلات TSNE نهایی این حالت به شکل زیر است:

روش دیگر که به کار گرفتیم ایجاد یک underclustring است و سپس کلاستر کردن مجدد کلاستر بزرگتر است که ممکن است شامل زیر بخش باشد. ابتدا kmeans با k برابر با ۳ بر روی داده ها اجرا کنیم، سپس با استفاده agglomerative کلاسترینگ، دوباره کلاستری بندی میکنیم. کلاستر بزرگتر را که با استفاده از TSNE پلات کنیم، به شکل زیر میرسیم:



باز هم در این مدل مقداری تداخل کلاستری وجود دارد.

در نهایت مدلی که با الگوریتم های سلسله مراتبی ساخته شده بود، با تعداد کلاستر ۵ ، (صفحه ۱۰) نتایج بهتری را نمایش میداد. حال این مدل را بر روی داده های تست بدون لیبل تست میکنیم.



مدل های بدون نیاز به لیبل معمول نیز نتایج زیر را میدهند:

```
print(calinski_harabasz_score(test_features, predict))
print(silhouette_score(test_features, predict))
print(davies_bouldin_score(test_features, predict))
```

```
2363.7299617895596
0.15562463
2.0927549700638766
```

در قسمت های قبلی توضیح داده شد چرا ممکن است برخی از این متریک ها امتیاز مناسبی ندهند.

در گام آخر نیز تست بر روی داده های دارای لیبل مطرح میشود. از جهت مپ کردن خوشه ها با دامین ها، بصورت one to one لازم است انجام شود.

نتیجه متریک ها بر روی داده های دارای لیبل دامین به قرار زیر است:

```
score = adjusted_mutual_info_score(image_domain,predict_domain)

print(score)
```

```
0.8294122888189421
```

```
score = adjusted_rand_score(image_domain,predict_domain)

print(score)
```

```
0.8060650170337942
```

نتیجه نسبتا مطلوبی بدست می آید.