مبانی هوش محاسباتی (پروژه اول)

کیارش آستان‌بوس ۴۰۰۱۲۶۲۵۷۰
محمدرضا نادری ۴۰۰۱۲۶۲۳۸۶

مهر ۱۴۰۳

فهرست مطالب

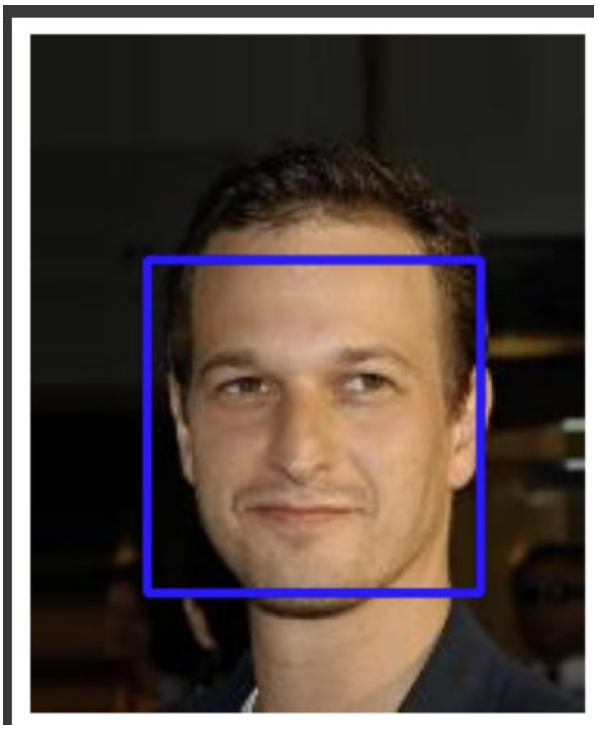
۱) فاز اول

برای گرفتن شناسایی چهره از کتابخانه OpenCV استفاده شده است. برای این منظور یک شی از نوع cascadeClassifier رو ایجاد کردیم که برای شناسایی چهره استفاده میشود همچنین از فایل xml به نام haarcascade_frontalface_default.xml استفاده شده است که مدل های آماده ای را برای شناسایی چهره ها دارد که با استفاده از الگوریتم های ML آموزش داده شده اند و میتوانند چهره ها را از تصویر شناسایی کند.

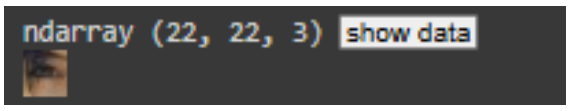
پس از تشخیص چهره با استفاده ازتابع get_skin_color میانگینی از رنگ چهره های شناسایی شده را محاسبه میکنیم

برای تشخیص چشم در تصویر با توجه به مختصاتی که از OpenCV دریافت کردیم که شامل (x، y، w h) است ضریبی را به این مقادیر اضافه میکنیم تا مختصات چشم را به ما بدهد که برای بدست آوردن مختصات ضرایب گوناگون را تست میکنیم با توجه به این مختصات بدست آمده مجدد عکس به همراه مختصات را به تابع get_eye_color .میدهیم تا میانگین رنگ چشم را به ما بدهد

برای چک کردن مختصات دستی که برای چشم بدست آوردیم. مجدد با استفاده از کتابخانه openCv .مختصات چشم را بدست آوردیم که این مختصات بدست آمده بسیار نزدیک به مختصاتی بود که به صورت دستی بدست آوردیم



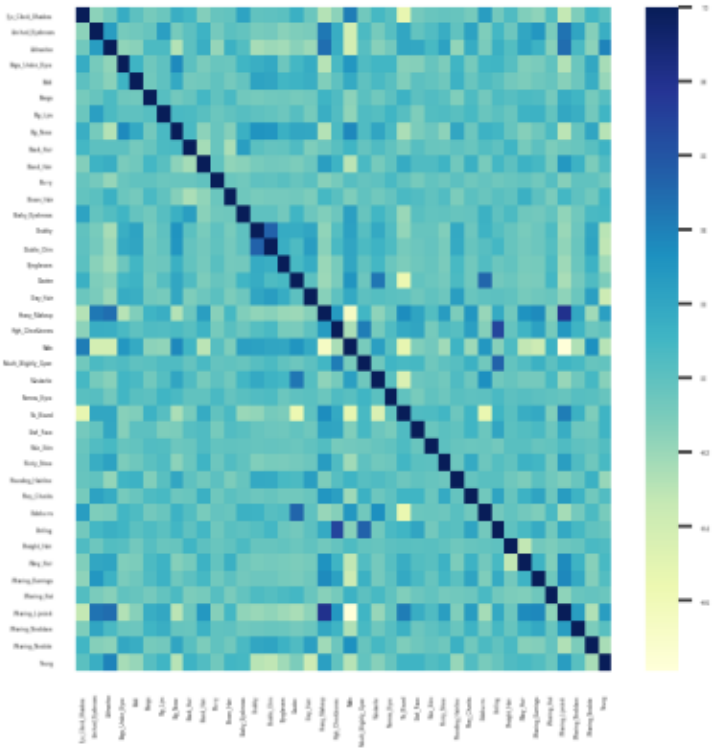
شکل ۱: اثبات درستی مختصات صورت



شکل ۲: اثبات درستی مختصات چشم

۲) فاز دوم

در این فاز با استفاده از correlation Pearson ،coefficient ماتریس همبستگی را حساب کرده و فیچر های مناسب را انتخاب میکنیم



شکل ۳: correlation matrix heatmap

بعد از تشکیل ماتریس همبستگی با استفاده از تابع select_features_by_correlation فیچر هارا انتخاب میکنیم. این تابع به صورت حریصانه فیچر هایی راانتخاب می‌کند که با فیچر های انتخاب شده همبستگی کمتر از threshhold داشته باشد، همبستگی کمتر از threshhold داشته باشند.

5_o_Clock_Shadow Bald Big_Lips Blurry Pale_Skin Pointy_Nose

شکل ۴: ویژگی های انتخاب شده

threshhold = ۰.۰۶

البته انتخاب فیچر ها فقط با استفاده از همبستگی ممکن است بهینه نباشد. مثلا اگر واریانس فیچری پایین باشد اطلاعات زیادی برای ما ممکن است تولید نکند یا با استفاده از روش هایی مانند PCA با ترکیب فیچر ها، فیچر های جدید تولید کنیم.

	5_o_Clock_Shadow	Bald	Big_Lips	Blurry	Pale_Skin	Pointy_Nose
5_o_Clock_Shadow	1.000000	0.007525	-0.044157	-0.031342	-0.038736	-0.024447
Bald	0.007525	1.000000	0.000379	-0.009016	-0.018288	-0.057184
Big_Lips	-0.044157	0.000379	1.000000	-0.041550	0.039522	0.052071
Blurry	-0.031342	-0.009016	-0.041550	1.000000	-0.016046	-0.052030
Pale_Skin	-0.038736	-0.018288	0.039522	-0.016046	1.000000	0.007182
Pointy_Nose	-0.024447	-0.057184	0.052071	-0.052030	0.007182	1.000000

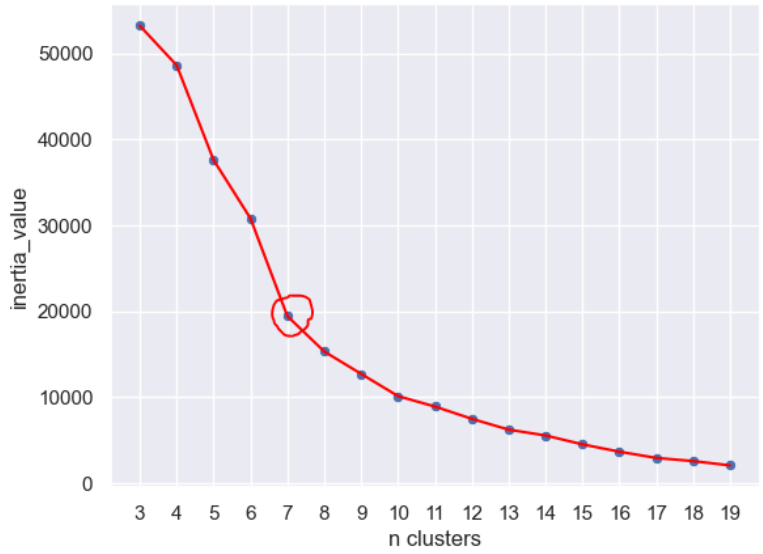
شکل ۵: همبستگی ویژگی های انتخاب شده

۳) فاز سوم

۱.(۳ K-Means

۱.۱.(۳ انتخاب بهترین K

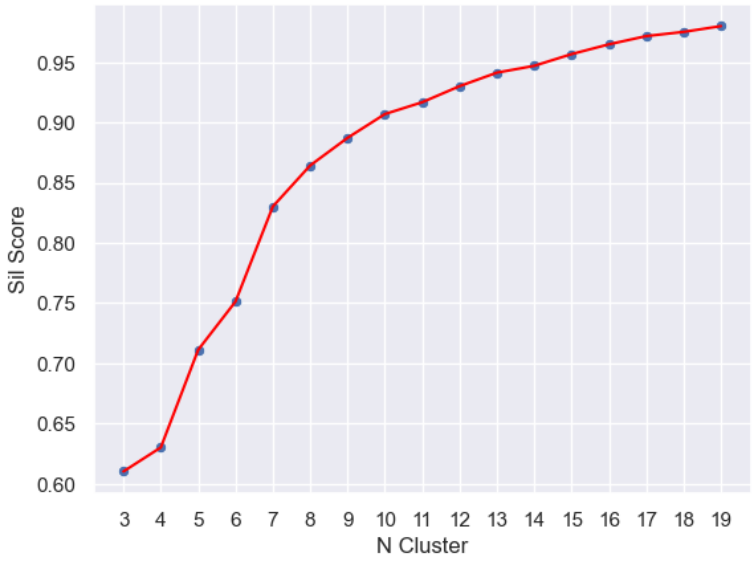
در این قسمت با استفاده از دو روش Lbow Method و Silhouette Score بهترین K را محاسبه خواهیم کرد.



شکل ۶: Lbow Method

Lbow Method مجموع فواصل مربعی بین هر نقطه و مرکز خوشه‌اش را محاسبه می‌کند. این معیار نشان‌دهنده میزان پراکندگی نقاط در خوشه‌ها است؛ هرچه این مقدار کمتر باشد، خوشه‌ها متراکم‌تر هستن برای انتخاب بهترین K باید نقطه ای

را در نمودار انتخاب کنیم که بیشترین شکستگی در آن اتفاق می‌افتد و بعد از آن مقدار فاصله با شیب کمتری کم می‌شود . همانطور که در شکل ۳ مشخص است، بهترین مقدار K برابر ۷ است



شکل ۷: Silhouette Score

معیار سیلوئت کیفیت خوشه‌بندی را بر اساس نزدیکی نقاط به خوشه خود و دوری از خوشه‌های دیگر ارزیابی می‌کند. مقدار سیلوئت بین −۱ تا +۱ است؛ هرچه مقدار به +۱ نزدیک‌تر باشد، کیفیت خوشه‌بندی بهتر است. این مقدار به طور طبیعی با افزایش تعداد خوشه ها به یک نزدیک تر می‌شود اما در K = ۷ مشاهده می‌شود که شیب بالا رفتن این مقدار کاهش می‌یابد.

۲.۱.(۳ HyperParameter Tuning

در این قسمت، الگوریتم K-Means را با مقادیر مختلف پارامتر ها اجرا می‌کنیم و بر اساس معیار های ارزیابی، بهترین مقادیر برای هر پارامتر را انتخاب می‌کنیم. باید توجه شود که مقدار random_state را عدد ثابتی باید بگذاریم که در هر دفعه اجرا، نتیجه ها یکسان باشد و از شانسی بودن seed اولیه تاثیر نپذیرد.

	K	init	n_init	max_iter	algorithm	Inertia	Silhouette Score	Davies-Bouldin Index	Calinski-Harabasz Index
0	7	k-means++	10	300	lloyd	19399.790434	0.830199	0.521425	42623.328429
1	7	k-means++	10	300	elkan	19654.757855	0.833730	0.423287	41962.318446
2	7	k-means++	10	500	lloyd	19399.790434	0.830199	0.521425	42623.328429
3	7	k-means++	10	500	elkan	19654.757855	0.833730	0.423287	41962.318446
4	7	k-means++	20	300	lloyd	19399.790434	0.830199	0.521425	42623.328429
5	7	k-means++	20	300	elkan	19479.495016	0.835655	0.443841	42414.832962
6	7	k-means++	20	500	lloyd	19399.790434	0.830199	0.521425	42623.328429
7	7	k-means++	20	500	elkan	19479.495016	0.835655	0.443841	42414.832962
8	7	random	10	300	lloyd	23919.397944	0.802293	0.572058	32995.208239
9	7	random	10	300	elkan	23390.309740	0.807430	0.564116	33930.032394
10	7	random	10	500	lloyd	23919.397944	0.802293	0.572058	32995.208239
11	7	random	10	500	elkan	23390.309740	0.807430	0.564116	33930.032394
12	7	random	20	300	lloyd	23919.397944	0.802293	0.572058	32995.208239
13	7	random	20	300	elkan	23390.309740	0.807430	0.564116	33930.032394
14	7	random	20	500	lloyd	23919.397944	0.802293	0.572058	32995.208239
15	7	random	20	500	elkan	23390.309740	0.807430	0.564116	33930.032394

شکل ۸: HyperParameter Tuning

با توجه به شکل ۵ ، جز الگوریتم، تفاوت چندانی در پارامتر ها وجود ندارد و پارامتر های نهایی را به صورت زیر انتخاب می‌کنیم

- **init** : K-means++
- **n_init** :بدون تاثیر چشم گیر (defualt)
- **max_iter** : بدون تاثیر (defualt)
- **algorithm** : lloyd

Silhouette Score	Davies-Bouldin Index	Calinski-Harabasz Index	Inertia
0.83023	0.531267	42595.932046	19410.226404

شکل ۹: نتیجه نهایی

۳.۱.(۳ خوشه بندی با استفاده از رنگ پوست و چشم

	Silhouette Score	Davies-Bouldin Index	Calinski-Harabasz Index	Inertia
0	0.384228	0.8612	36539.826136	1.031264e+07

شکل ۱۰: اضافه کردن رنگ پوست و چشم

همانطور که در نتیجه معیار های ارزیابی مشخص است افت شدید کیفیت خوشه بندی را بعد از اضافه کردن رنگ چشم و پوست داریم. از دلایل این اتفاق میتوان به موارد زیر اشاره کرد

- پایین بودن کیفیت تصاویر
- بهینه نبودن روش انتخاب رنگ چشم. در روش استفاده شده ما ناحیه مستطیلی دور چشم را انتخاب کرده و میانگین رنگ های موجود را به عنوان رنگ چشم بر میگردانیم اما در اصل باید خود ناحیه چشم مشخص شود و پوستی همراهش نباشد.
- با توجه به ماهیت داده ،train بعضی عکس ها رنگ چشم ندارند و یا مشخص نشده‌اند. برای حل این مشکل میانگین کل ستون را برای مقادیر NaN قرار دادیم که این کار نیز در نتیجه خوشه بندی اثر منفی میگذارد
- میانگین گرفتن روی RGB ممکن است دو رنگ متفاوت باعدد های یکسان وجود داشته باشند مثلا (۱۰۰،۵۰،۲۰۰) و (۲۰۰ ، ۱۰۰ ، ۵۰) اما چون روی این مقادیر میانگین می‌گیریم که به یک ویژگی تبدیل شوند، رنگ های متفاوت در یک گروه قرار می‌گیرند.
- ممکن است با اضافه کردن ۲ تا ویژگی جدید، تعداد ویژگی ها از تعداد بهینه بیشتر شود و باید ویژگی دیگری را در عوض حذف می‌کردیم.

۲.(۳ DBSCAN

این الگوریتم در کل یک الگوریتم خوشه بندی است که ساختار داده را حفظ میکند . در این الگوریتم دو پارامتر اصلی eps و min_sample وجود دارد

eps (epsilon) : این پارامتر حداکثر فاصله‌ای را تعیین می‌کند که دو نقطه باید از هم فاصله داشته باشند تا به عنوان همسایه در نظر گرفته شوند. اگر فاصله بین دو نقطه کمتر از eps باشد، آن‌ها به عنوان همسایه‌های نزدیک در نظر گرفته می‌شوند.

min_samples : این پارامتر حداقل تعداد نقاطی را تعیین می‌کند که باید در نزدیکی یک نقطه (در دایره‌ای به شعاع eps) وجود داشته باشد تا آن نقطه به عنوان یک هسته (core point) شناخته شود. اگر یک نقطه هسته باشد، می‌تواند نقاط دیگر را به خوشه اضافه کند.

	eps	min_samples	silhouette_score	davies_bouldin_score	calinski_harabasz_score	clusters
0	0.5	5	0.999420	0.907651	906910.393778	37
1	0.5	12	0.998333	0.957590	345011.393885	33
2	0.5	18	0.996592	1.010839	192564.831284	29
3	0.6	5	0.999420	0.907651	906910.393778	37
4	0.6	12	0.998333	0.957590	345011.393885	33
5	0.6	18	0.996592	1.010839	192564.831284	29
6	0.7	5	0.999420	0.907651	906910.393778	37
7	0.7	12	0.998333	0.957590	345011.393885	33
8	0.7	18	0.996592	1.010839	192564.831284	29
9	0.8	5	0.999420	0.907651	906910.393778	37
10	0.8	12	0.998333	0.957590	345011.393885	33
11	0.8	18	0.996592	1.010839	192564.831284	29
12	0.9	5	0.999420	0.907651	906910.393778	37
13	0.9	12	0.998333	0.957590	345011.393885	33
14	0.9	18	0.996592	1.010839	192564.831284	29

شکل ۱۱: HyperParameter Tuning

با توجه به نتایج بدست آمده اگر مقدار پارامتر eps را کوچک در نظر بگیریم باعث شناسایی خوشه های کوچک و پراکنده میشود . و بلعکس اگر eps را بزرگ در نظر بگیریم باعث ایجاد خوشه ها بزرک تر میشود که داده های نویز زیادی را در بر میگیرد

همچنین برای پارامتر min_samples اگر مقدار کمی را در نظر بگیریم طبق نتایج باعث ایجاد خوشه کوچک تر و تعداد بیشتری از نقاط به عنوان هسته میشود اما اگر مقدار زیادی به این پارامتر بدهیم باعث ایجاد خوشه های بزرگ تری میشود چون نقاطی را به عنوان هسته انتخاب میکند که تراکم زیادی دارد

با توجه به نتایج بدست آمده و اینکه به صورت کلی مقدار min_sample اگر دوبرابر تعداد پارامتر های انتخابی باشد نتیجه معقولی را میدهد. مقدار پارامتر هارا به صورت زیر در نظر گرفتیم که با توجه به ارزیابی ها، نتیجه نسبتا خوبی می‌دهد.

- **eps** : ۰.۵

- **min_samples** :۱۲ (دو برابر تعداد فیچر ها)

	Silhouette Score	Davies-Bouldin Index	Calinski-Harabasz Index	clusters
0	0.99986	4.689630e-08	3.322264e+29	33

شکل ۱۲: نتیجه نهایی

MeanShift ۳.(۳

الگوریتم meanshift یک روش خوشه بندی غیرپارامتریک است که هدف آن شناسایی مناطق دارای چگالی بالا است

این الگوریتم یک پارامتر bandwidth دارد که تعیین‌کننده‌ی شعاع یا اندازه‌ی ناحیه‌ای است که برای محاسبه‌ی میانگین استفاده می‌شود.

با توجه به نتایج بدست آمده دادن مقدار کوچک به bandwidth باعث شناسایی ساختار های پیچیده تر و دقیق تر میشود اما ممکن است باعث شناسایی خوشه های زیادی شود همچنین اگر این مقدار را زیاد در نظر بگیریم باعث ایجاد خوشه های بزرگ تر و متراکم تر میشود اما ممکن است خوشه های کوچک و نقاط پراکنده را به هم متصل کند و باعث ایجاد خوشه های کمتری شود

	bw	silhouette_score	davies_bouldin_score	calinski_harabasz_score	clusters
0	0.100000	0.99986	4.689630e-08	3.322264e+29	47
1	0.255556	0.99986	4.689630e-08	3.322264e+29	47
2	0.411111	0.99986	4.689630e-08	3.322264e+29	47
3	0.566667	0.99986	4.689630e-08	3.322264e+29	47
4	0.722222	0.99986	4.689630e-08	3.322264e+29	47
5	0.877778	0.99986	4.689630e-08	3.322264e+29	47
6	1.033333	0.99986	4.689630e-08	3.322264e+29	47
7	1.188889	0.99986	4.689630e-08	3.322264e+29	47
8	1.344444	0.99986	4.689630e-08	3.322264e+29	47
9	1.500000	0.99986	4.689630e-08	3.322264e+29	47

شکل ۱۳: eps ۰.۱ to ۱.۵

	bw	silhouette_score	davies_bouldin_score	calinski_harabasz_score	clusters
0	2.000	0.999860	4.689630e-08	3.322264e+29	47
1	2.125	0.999860	4.689630e-08	3.322264e+29	47
2	2.250	0.681856	5.471417e-01	1.054172e+04	25
3	2.375	0.592920	6.503576e-01	7.766572e+03	14
4	2.500	0.592920	6.503576e-01	7.766572e+03	14

شکل ۱۴: ۲ to ۲.۵ eps

با توجه به جداول بالا می‌توانیم نتایج زیر را بگیریم:

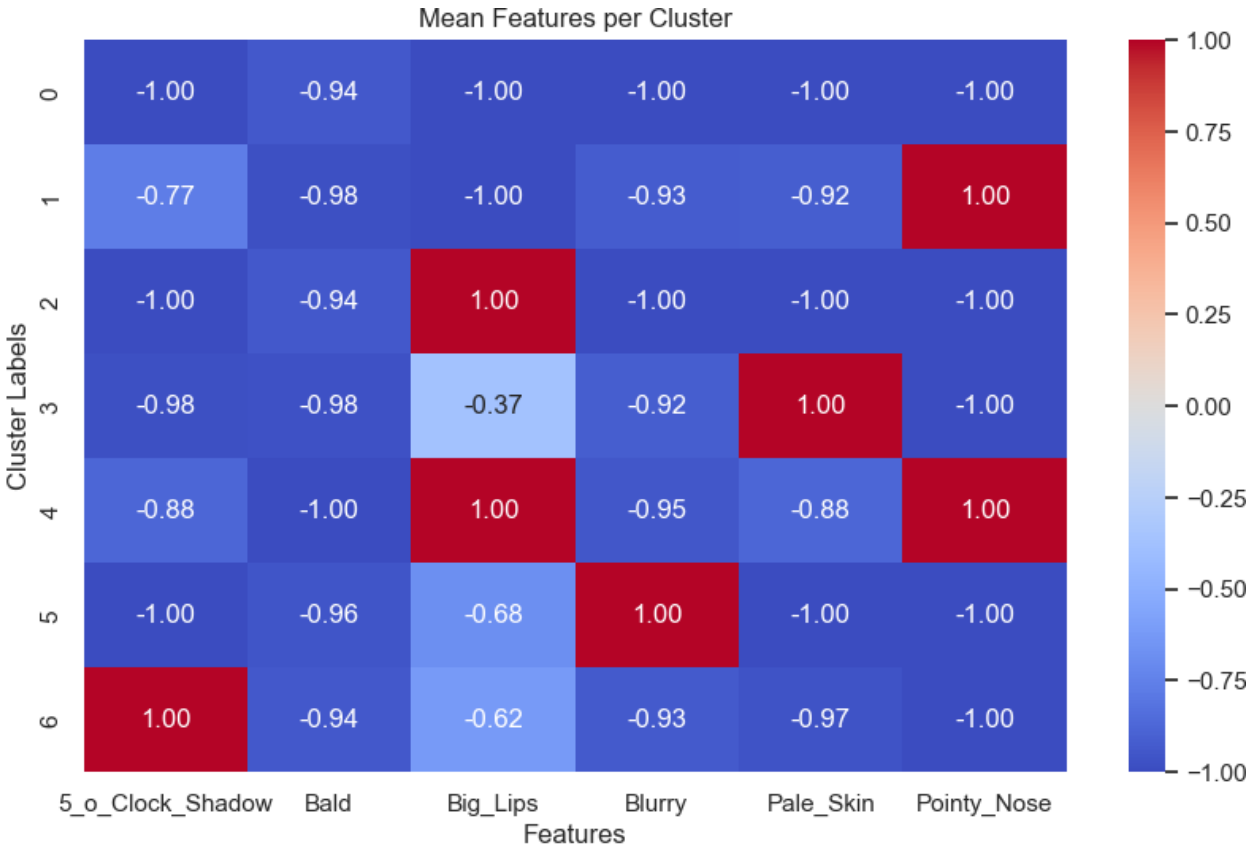
- **ثبات در خوشه‌بندی با bandwidth پایین (۱.۰ تا ۱۲۵.۲):** : زمانی که پهنای باند از ۱۲۵.۲ فراتر می‌رود، خوشه‌ها شروع به ادغام شدن می‌کنند. bandwidth های بالا باعث می‌شوند که الگوریتم MeanShift نواحی بیشتری را به عنوان یک خوشه ببیند و قله‌های نزدیک به هم را به عنوان یک خوشه واحد تشخیص دهد. این رفتار در روش‌های مبتنی بر تراکم رایج است؛ bandwidth های بزرگ‌تر، تفاوت‌های تراکم محلی را محو می‌کنند و خوشه‌های متمایز را ادغام می‌کنند.
- **کاهش تعداد خوشه‌ها با افزایش پهنای باند:** زمانی که پهنای باند از ۱۲۵.۲ فراتر می‌رود، خوشه‌ها شروع به ادغام شدن می‌کنند. bandwidth های بالا باعث می‌شوند که الگوریتم MeanShift نواحی بیشتری را به عنوان یک خوشه ببیند و قله‌های نزدیک به هم را به عنوان یک خوشه واحد تشخیص دهد. این رفتار در روش‌های مبتنی بر تراکم رایج است؛ bandwidth های بزرگ‌تر، تفاوت‌های تراکم محلی را محو می‌کنند و خوشه‌های متمایز را ادغام می‌کنند.

با توجه به مقادیر و ارزیابی جدول شکل ۱۲ و ارزیابی انجام شده و همچنین با استفاده از تابع estimate_bandwidth که مربوط به کتابخانه sklearn و مقدار مطلوب bandwidth را میدهد مقدار ۱۸۶۰۲۲.۱ را برای این پارامتر در نظر گرفتیم و با توجه به جدول تعداد ۴۷ cluster .را به ما میدهد

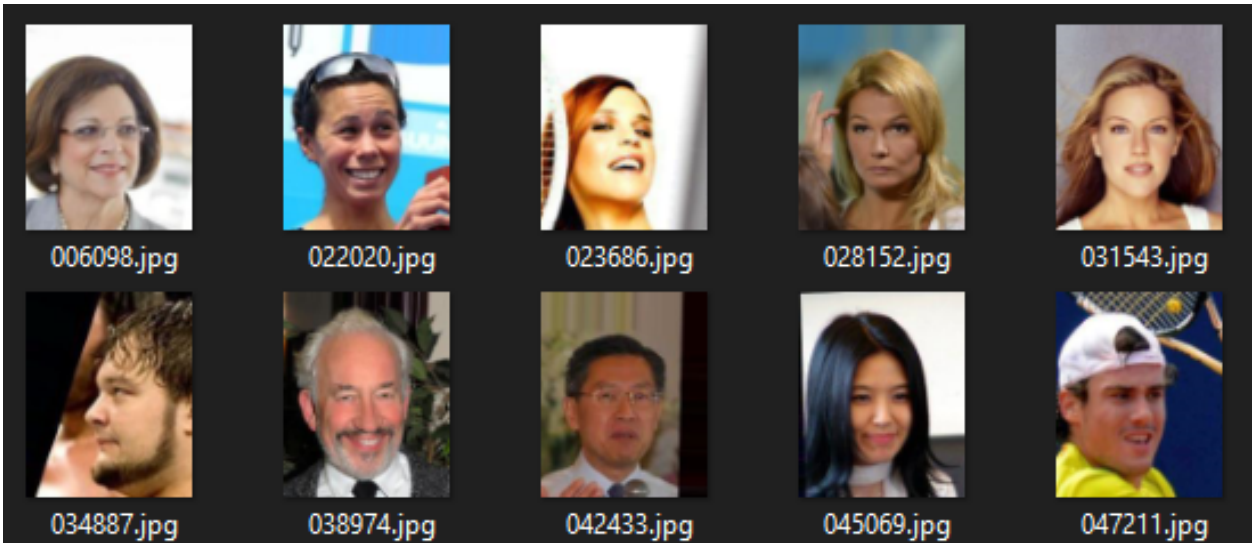
	Silhouette Score	Davies-Bouldin Index	Calinski-Harabasz Index	clusters
0	0.99986	4.689630e-08	3.322264e+29	47

شکل ۱۵: نتیجه نهایی

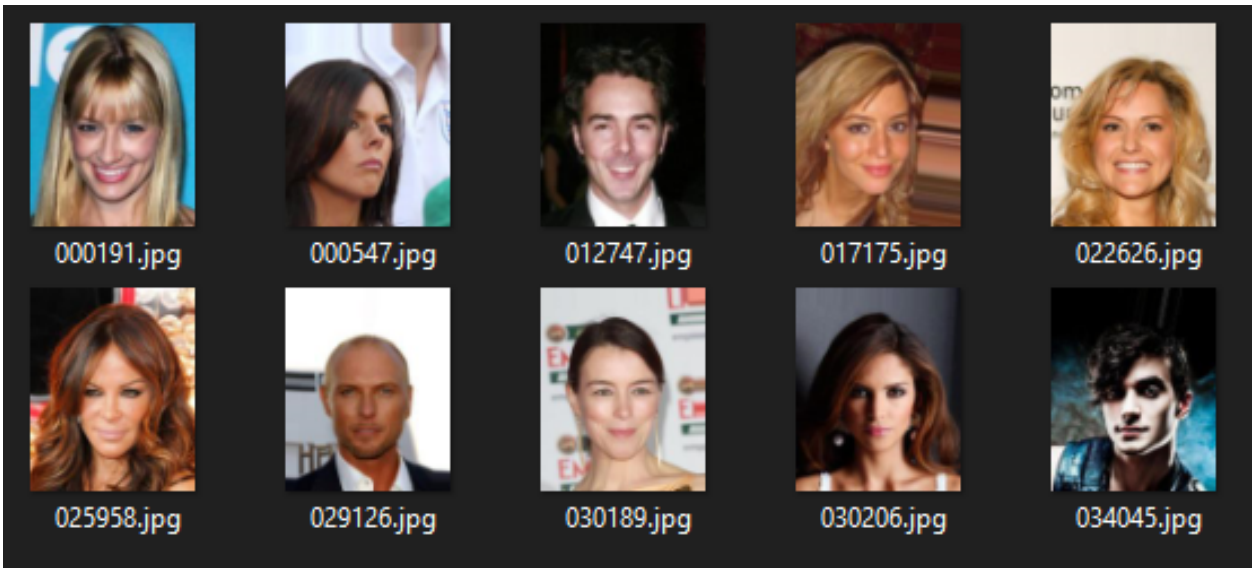
۴.(۳ و خوشه بندی ۱۰ عکس HeatMap



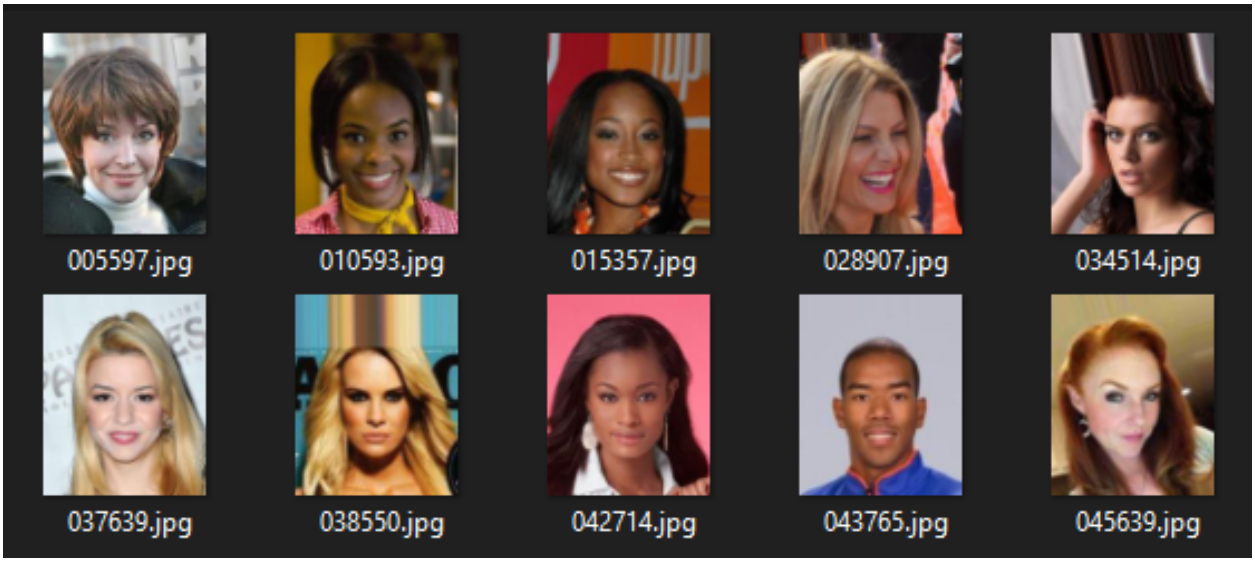
شکل ۱۶: Kmeans Heatmap



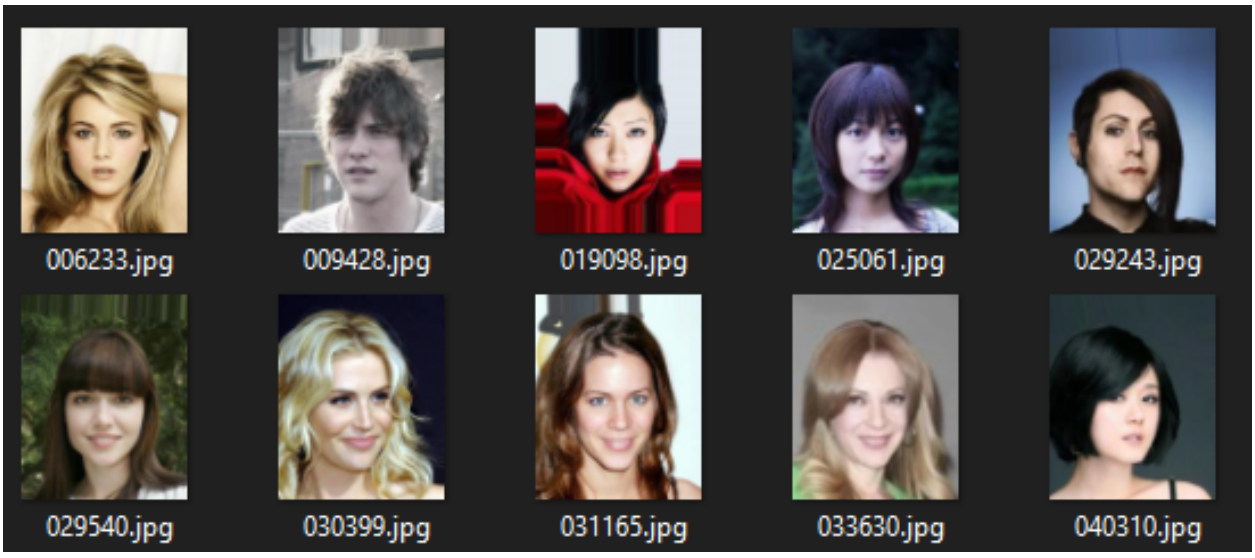
شکل ۱۷: cluster ۰



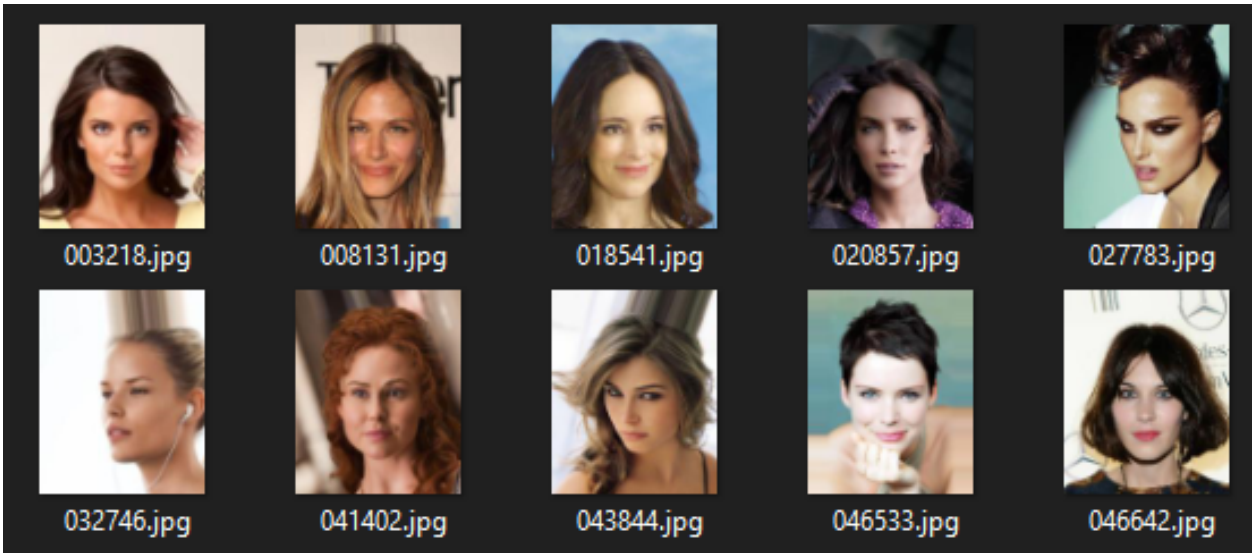
شکل ۱۸: cluster ۱



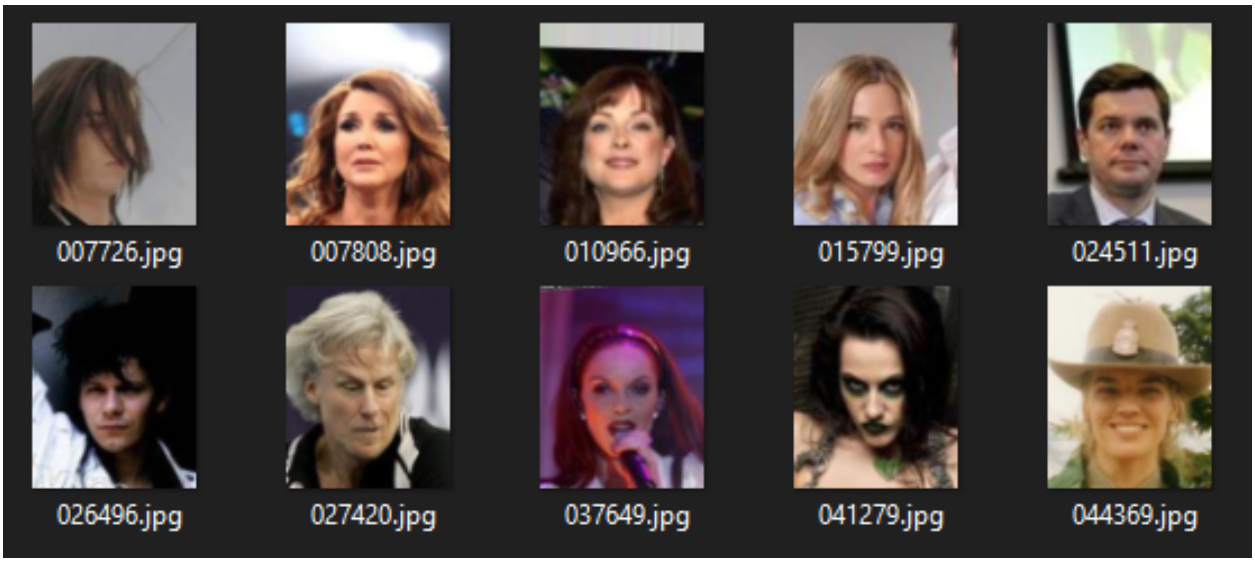
شکل ۱۹: cluster ۲



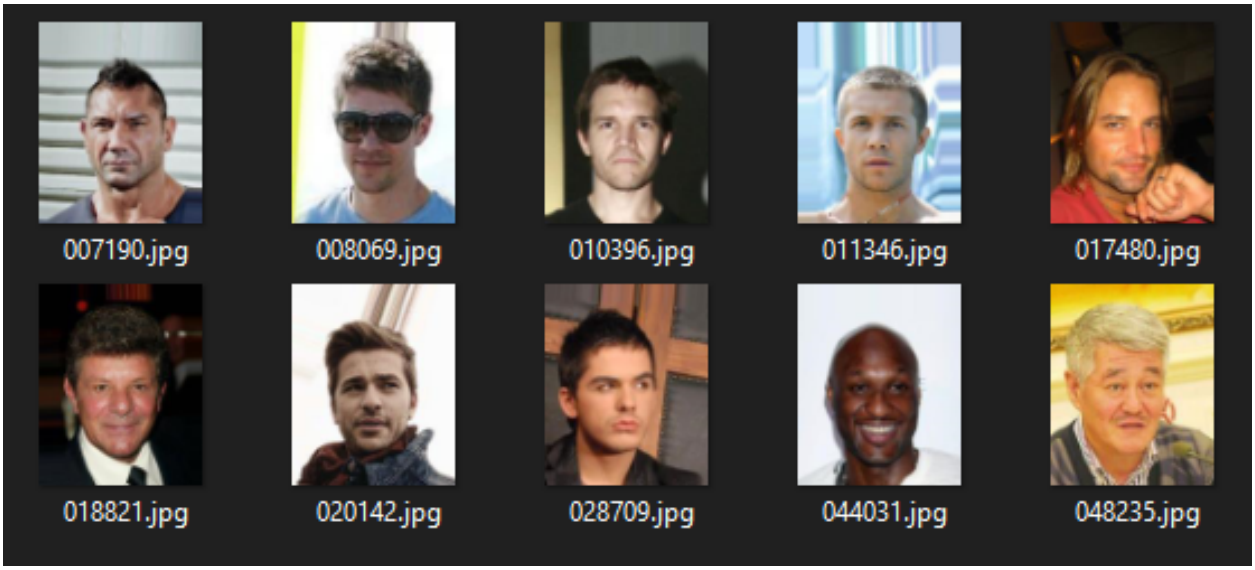
شکل ۲۰: cluster ۳



شکل ۲۱: cluster ۴



شکل ۲۲: cluster ۵



شکل ۲۳: ۶ cluster

با توجه به هیت مپ میتوان کلاستر هارا به صورت زیر ارزیابی کرد

- **cluster ۰ :** اکثرا مو دارند، بینی غیر نوک تیز، پوست غیر روشن و لب کوچک. این موارد به راحتی قابل بررسی در عکس های نمایش داده شده هستند و همچنین با فایل csv ویژگی ها مطابقت دارند. البته که بعضی موارد ممکن است نسبی باشد مانند روشنی پوست و شاید بتوان عکس ۰۰۶۰۹۸ را پوست روشن در نظر گرفت اما طبق csv کاملا درست است. (در csv پوست تیره ثبت شده)
- **cluster ۱ :** .این گروه همگی بینی نوک تیز دارند، اکثرا مو دارند و همگی لب های کوچک. ویژگی بارز این گروه بینی نوک تیز است در عکس ها به صورت واضح قابل مشاهده است
- **cluster ۲ :** ویژگی بارز این گروه لب های گنده است. با توجه به عکس های نمایش داده شده این موضوع را میتوان در اکثر عکس ها مخصوصا ۱۵۳۵۷ یا ۵۵۹۷ مشاهده کرد. شاید بعضی عکس هارا با توجه به وضوح پایین عکس، مشخص کرد ولی طبق csv تمام عکس های نمایش داده شده دارای لب های بزرگ هستند.
- **cluster ۳ :** ویژگی بارز این گروه پوست روشن و رنگ پریده است. طبق عکس ها میتوان این موضوع را به وضوح دید و برعکس بعضی خوشه‌های دیگر که افراد سیاه پوست در آنها بودند، این گروه هیچ سیاه پوستی ندارد
- **cluster ۴ :** ویژگی مهم این گروه داشتن بینی نوک تیز به همراه لب های بزرگ است. هر دو این ویژگی هارا میتوان به وضوح دید مخصوصاعکس ۸۱۳۱. شاید در بعضی عکس ها معلوم نباشد مانند ۴۱۴۰۲ ولی طبق csv کاملا این ویژگی هارا دارند.

- **cluster ۵ :** این گروه همگی دماغ غیر نوک تیز و دارای پوست معمولی هستند ویژگی بارز آنها Blurry است.
- **cluster ۶ :** ویژگی بارز این گروه داشتن ته ریش است. طبیعتا هیچ خانومی در این گروه وجود ندارد و همچنین طبق عکس ها همگی ته ریش دارند

۴) فاز چهارم

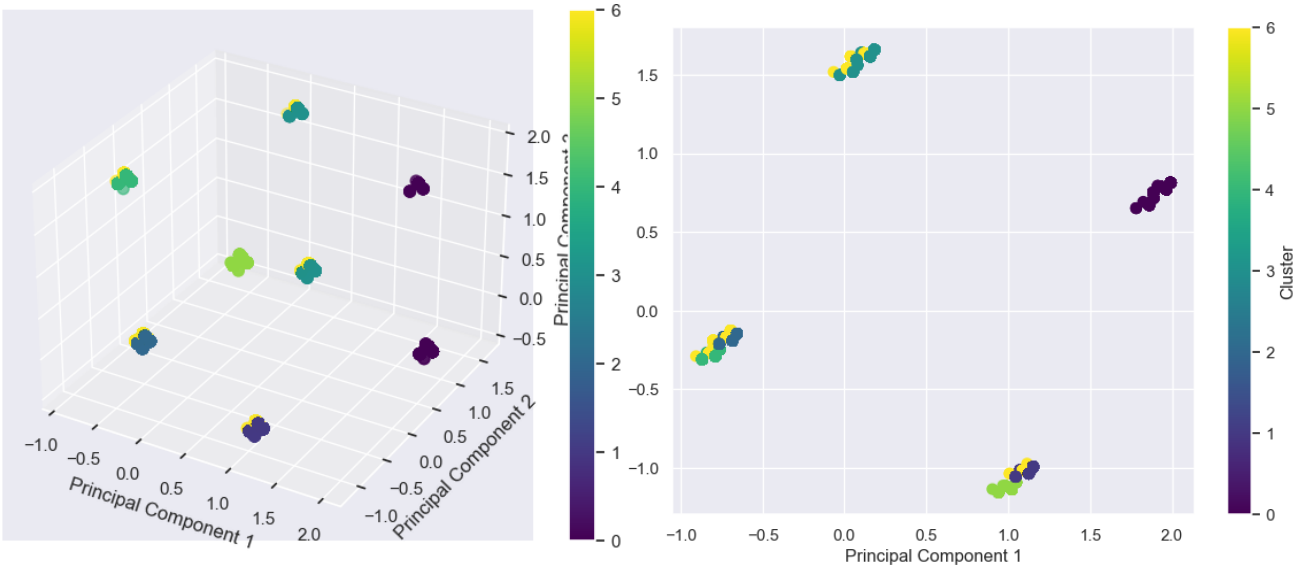
در این قسمت با استفاده از دو الگوریتم PCA و t-SNE ابعاد داده هارا کاهش داده و آن هارا نمایش می‌دهیم.

۱.(۴ PCA

Pca تکنیک آماری است که ابعاد داده را کاهش میدهد و هدف اصلی pca این هست که ویژگی اصلی و مهم داده ها را شناسایی میکند و با استفاده از آن ها ابعاد را کاهش میدهد در صورتی که بیشترین واریانس را حفظ میکند.

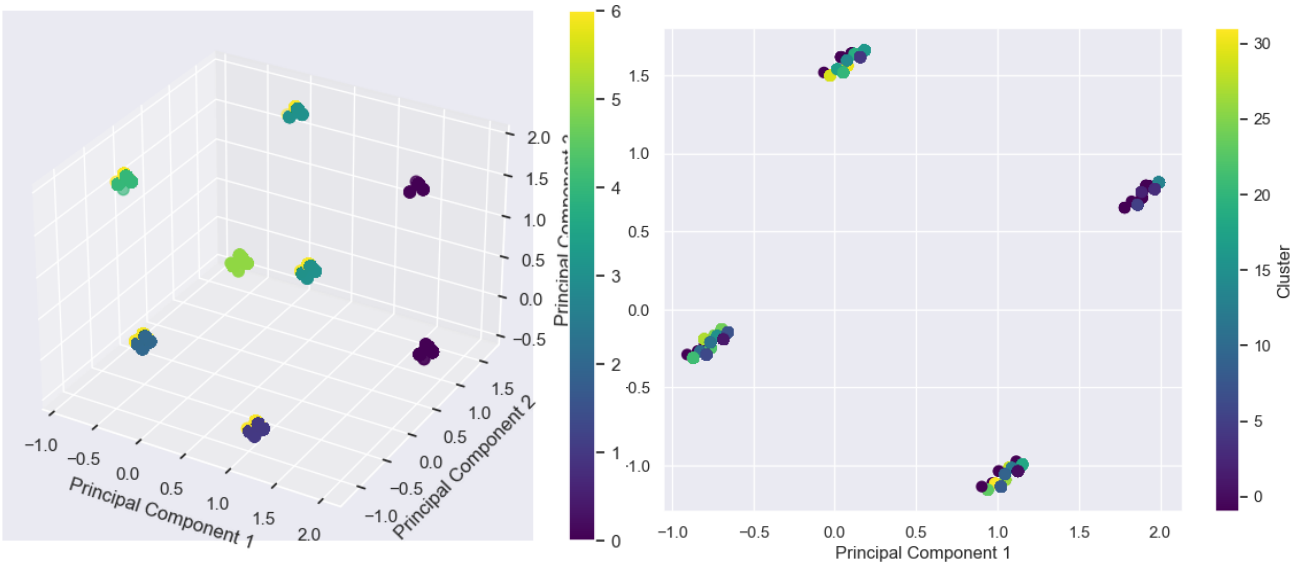
۱.۱.(۴ Kmeans

در شکل ۲۳، ۲۵، ۲۷ بدلیل اینکه به صورت ۲ بعدی نشان داده شده است داده ها رو هم افتاده و کمتر مشخص میشود که گروه بندی ها به چه صورت است اما در شکل ۲۴، ۲۶ و ۲۸ در ۳ بعد داده ها و گروه بندی آن ها به نمایش گذاشته شده است در نتیجه بهتر میتوان داده ها را بررسی کرد و داده هایی که روی هم افتاده است را میتوان تشخیص داد



شکل ۲۴: K-Means ۲D

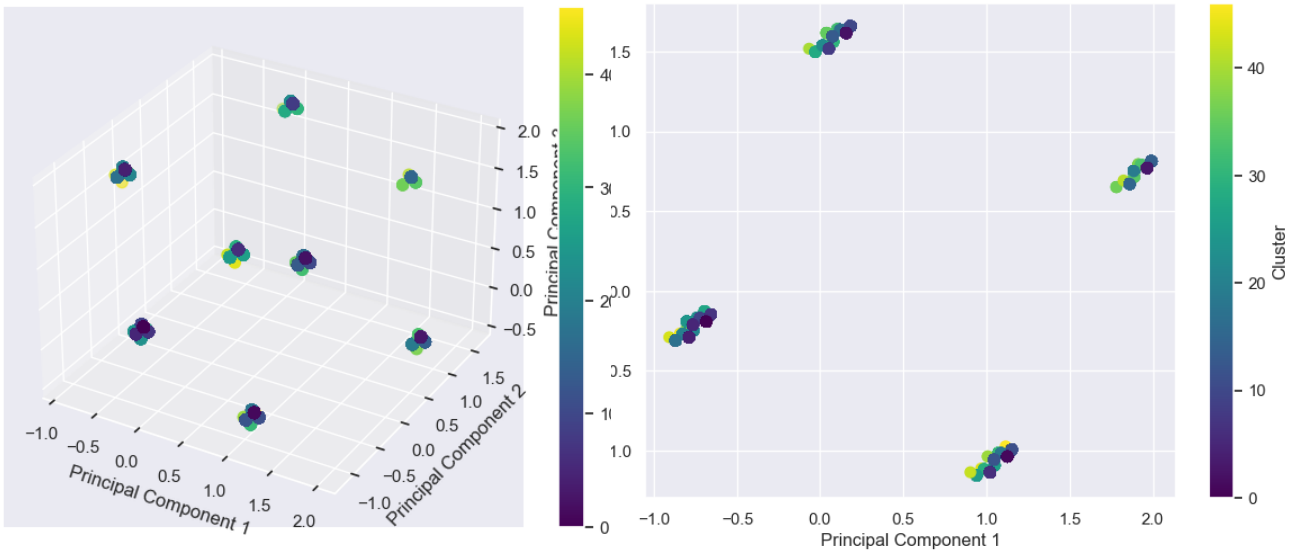
شکل ۲۵: K-Means ۳D



شکل ۲۶: DBSCAN ۲D

شکل ۲۷: DBSCAN ۳D

۲.۱.(۴ MeanShift



شکل ۲۸: MeanShit ۲D

شکل ۲۹: MeanShigy ۳D

t-SNE ۲.(۴

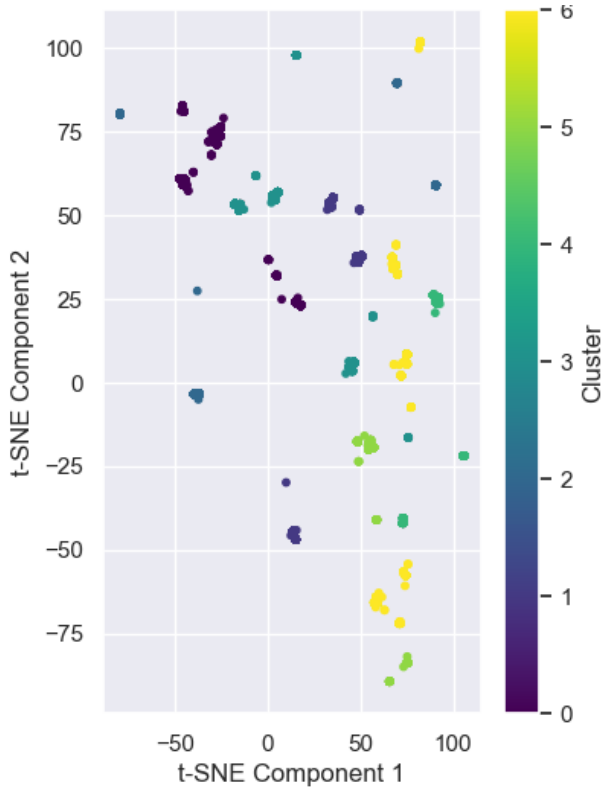
Tsne یک الگوریتم کاهش ابعاد است که برای تجسم داده ها طراحی شده است این الگوریتم داده های با ابعاد زیاد را به ۲ یا معمولا ۳ بعد تبدیل میکند به طوری که ساختار داده ها و روابط درون آن ها حفظ میشود

داده‌ها در خوشه‌های مجزا توزیع شده‌اند، اما برخی از خوشه‌ها نزدیک به یکدیگر قرار گرفته‌اند. این یعنی میتوانند این خوشه ها شباهت زیادی به یکدیگر داشته باشد

داده‌ها در خوشه‌های مجزا توزیع شده‌اند، اما برخی از خوشه‌ها نزدیک به یکدیگر قرار گرفته‌اند. این یعنی میتوانند این خوشه ها شباهت زیادی به یکدیگر داشته باشد

Kmeans ۱.۲.(۴

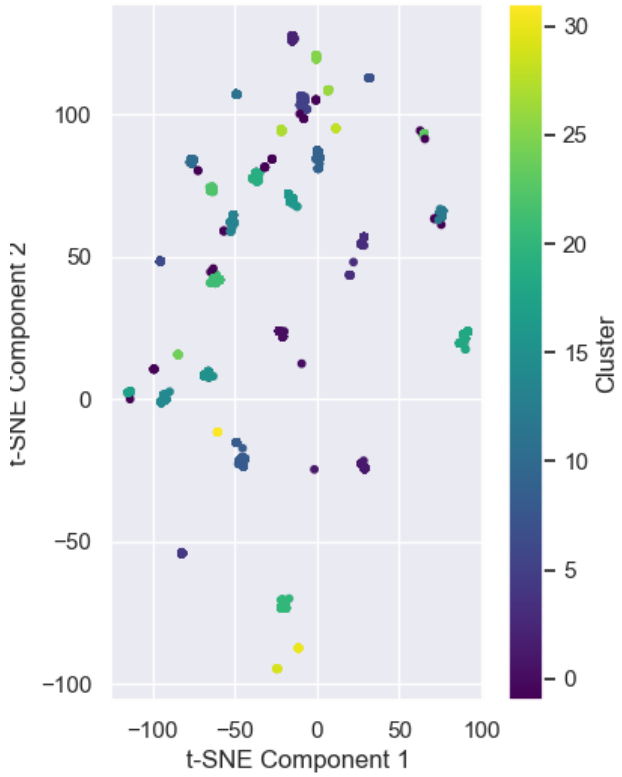
ممکن است خوشه‌ها را به شکل کروی در نظر بگیرد، بنابراین در t-SNE نقاط نزدیک به یکدیگر را به صورت مجموعه‌های فشرده نمایش می‌دهد



شکل ۳۰: K-Means ۲D

۲.۲.(۴ DBSCAN

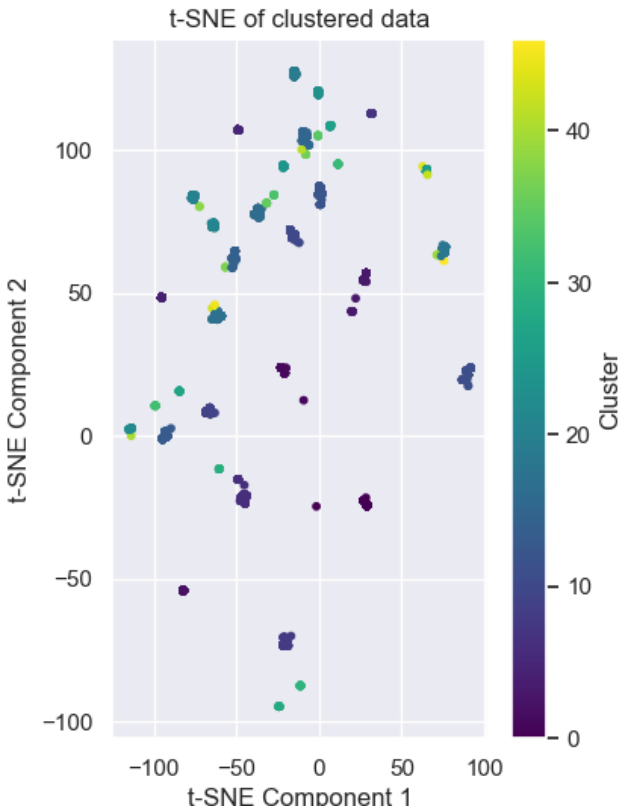
نقاط نویزی را جدا می‌کند و تنها نقاط متراکم را خوشه‌بندی می‌کند، که می‌تواند منجر به تشکیل خوشه‌هایی با شکل‌های غیر کروی در t-SNE شود.



شکل ۳۱: DBSCAN ۲D

می‌تواند خوشه‌های متراکم‌تر را نشان دهد، بنابراین در t-SNE نقاط متراکم ممکن است به خوشه‌هایی با مرزبندی طبیعی تبدیل شوند.

۳.۲.(۴ **MeanShift**



شکل ۳۲: K-Means ۲D

۵) فاز پنجم

```
kmeans
0    22029
1     9977
2     6609
6     4176
4     3792
5     1954
3     1463
Name: count, dtype: int64
```

شکل ۳۳: تعداد هر لیبل

	0	1	2	3	4	5	6
Centroid_0	50	0	0	0	0	0	0
Centroid_1	0	50	0	0	0	0	0
Centroid_2	0	0	50	0	0	0	0
Centroid_3	0	0	0	50	0	0	0
Centroid_4	0	0	0	0	50	0	0
Centroid_5	0	0	0	0	0	50	0
Centroid_6	0	0	0	0	0	0	50

شکل ۳۴: تست روی ۵۰ همسایه

	0	1	2	3	6	4	5
Centroid_0	3000	0	0	0	0	0	0
Centroid_1	0	3000	0	0	0	0	0
Centroid_2	0	0	3000	0	0	0	0
Centroid_3	1132	360	0	1440	68	0	0
Centroid_4	0	0	0	0	0	3000	0
Centroid_5	551	326	0	49	127	0	1947
Centroid_6	0	0	0	0	3000	0	0

شکل ۳۵: تست روی ۳۰۰۰ همسایه

با توجه به نتایج بدست آمده، در ۵۰ نزدیک ترین همسایه دقت صددرصد داریم ولی برای ۳۰۰۰ نزدیک ترین همسایه، همسایه های ۵ centroid به درستی مشخص شده اند ولی برای دو centroid دیگر،۷ نقطه از کلاستر ۵ در نزدیک ترین همسایه نبودند و ۲۳ نقطه از کلاستر ۳ در نزدیک ترین همسایه نبودند

دلایل این نتایج:

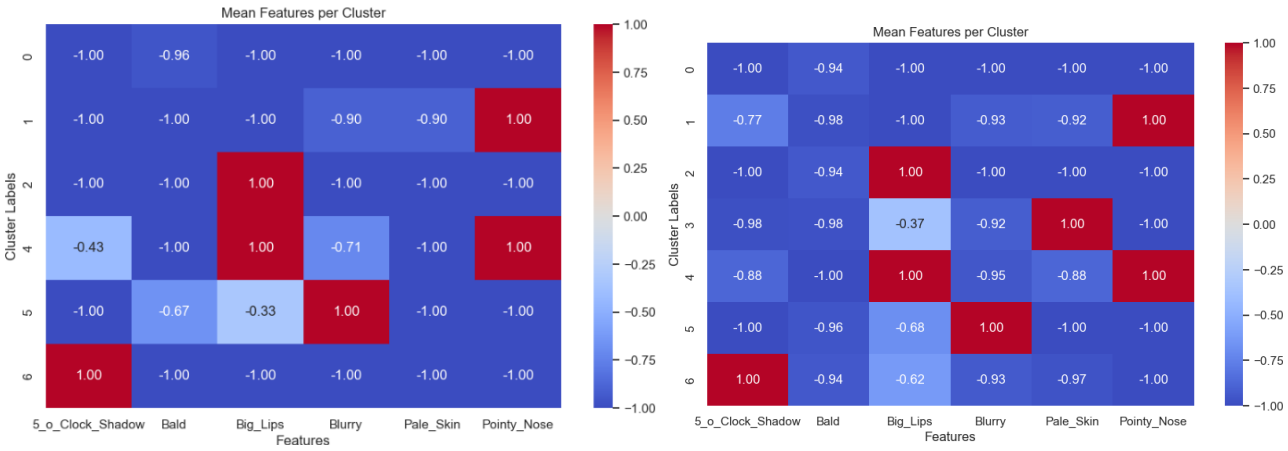
- **اثر فضای با ابعاد بالا:** : در فضاهای با ابعاد بالاتر، فاصله‌ها ممکن است کمتر تمایز داشته باشند (که به curse of dimensionality معروف است)، که باعث می‌شود KNN به سختی بتواند نقاطی را که واقعاً به یک خوشه تعلق دارند از

نقاط نزدیک در خوشه‌های دیگر تمیز دهد.

- **توزیع غیریکنواخت نقاط در کلاسترها:** اگر توزیع نقاط در کلاسترها غیریکنواخت باشد، امکان دارد نقاط خاصی از یک کلاستر در نزدیکی مرکز کلاسترهای دیگر قرار بگیرند و در تعداد بالای همسایگان نزدیک (مثل ۳۰۰۰) از مرزهای کلاستر خودشان خارج شوند.
- **تأثیر مقیاس بالای KNN:** در تعداد بالای همسایگان نزدیک، مانند ۳۰۰۰، نقاط بیشتری از لبه‌های کلاستر به نزدیکی مراکز کلاسترهای دیگر انتخاب می‌شوند. این وضعیت می‌تواند موجب شود نقاطی که باید در کلاستر خود باشند، در همسایگی کلاسترهای دیگر قرار بگیرند.

در کل نقاطی که توسط KNN مشخص نشده اند احتمالا نقاطی مرزی بوده اند و بجای آنها، به دلیل ابعاد زیاد نقاط مرزی کلاستر های دیگر انتخاب شده اند

۶) فاز ششم

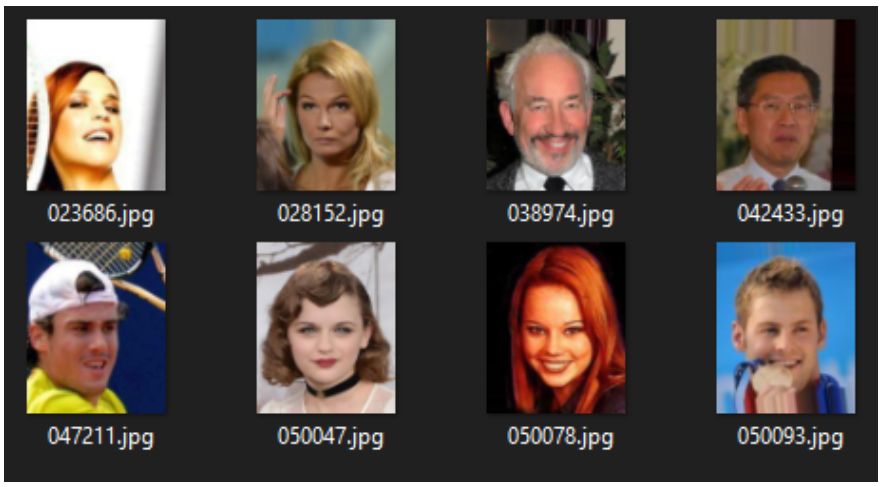


شکل ۳۶: Train HeatMap

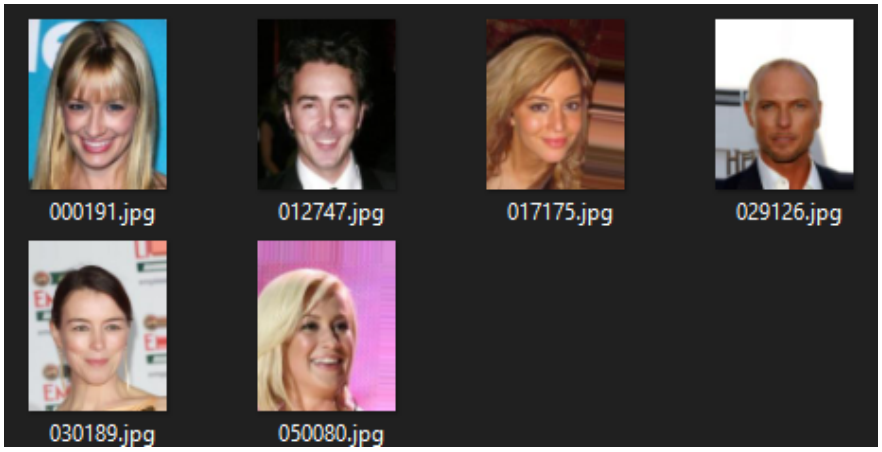
شکل ۳۷: Test HeatMap

	5_o_Clock_Shadow	Bald	Big_Lips	Blurry	Pale_Skin	Pointy_Nose	kmean	id
0	-1	-1	-1	-1	-1	-1	0	050078.jpg
1	-1	-1	-1	-1	-1	1	1	050080.jpg
2	-1	-1	1	-1	-1	-1	2	050012.jpg
3	-1	-1	1	-1	-1	1	4	050008.jpg
4	-1	-1	1	1	-1	-1	5	050054.jpg
5	1	-1	-1	-1	-1	-1	6	050048.jpg
6	-1	-1	-1	-1	-1	-1	0	050047.jpg
7	1	-1	-1	-1	-1	-1	6	050038.jpg
8	-1	-1	1	-1	-1	-1	2	050052.jpg
9	-1	-1	-1	-1	-1	-1	0	050093.jpg

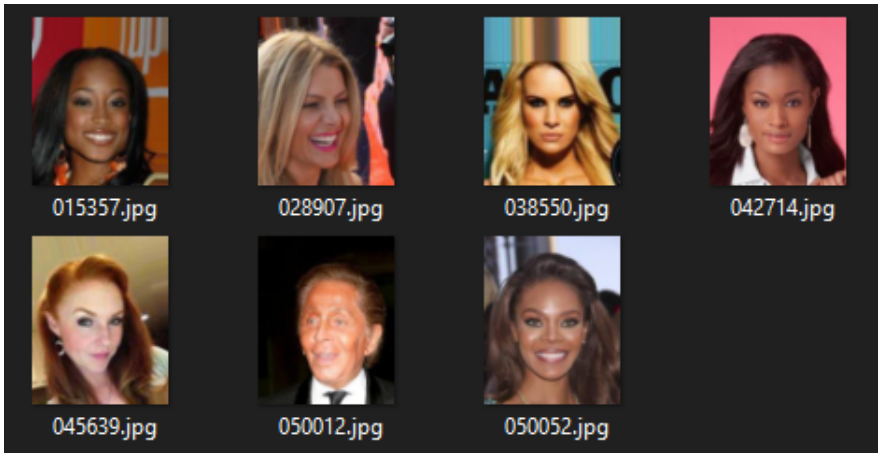
شکل ۳۸: ۱۰ داده انتخاب شده



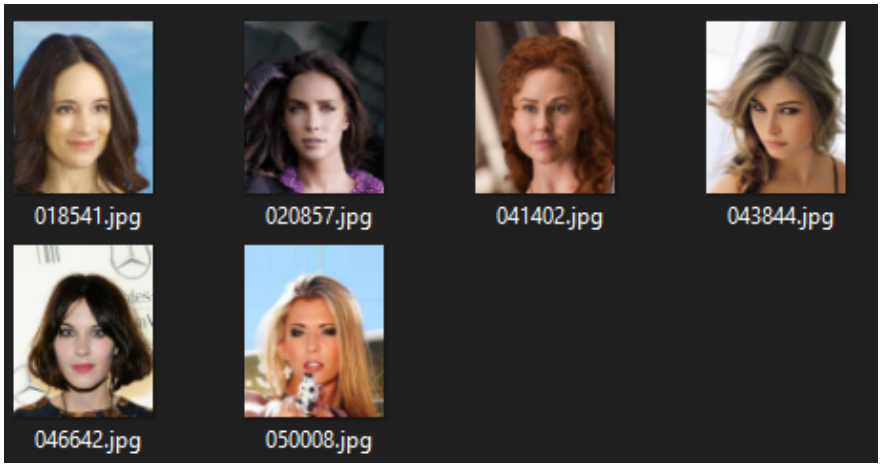
شکل ۳۹: cluster ۰



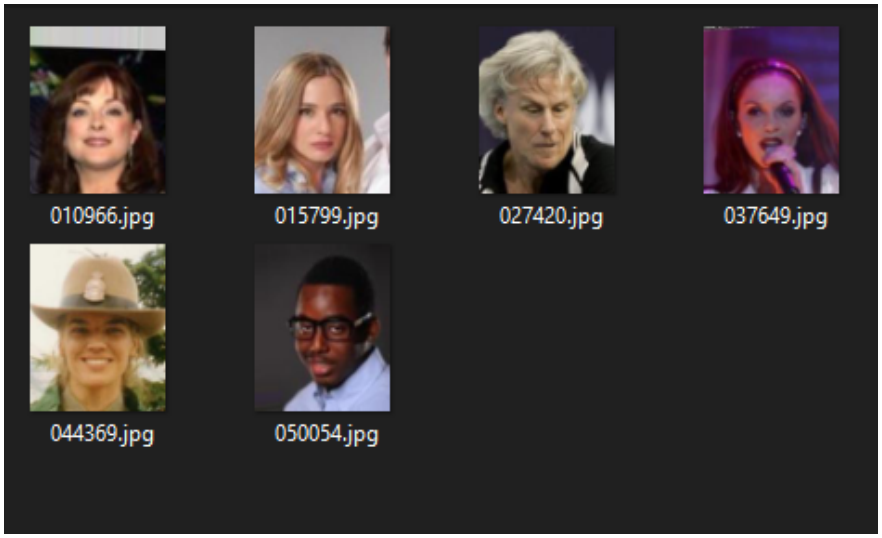
شکل ۴۰: cluster ۱



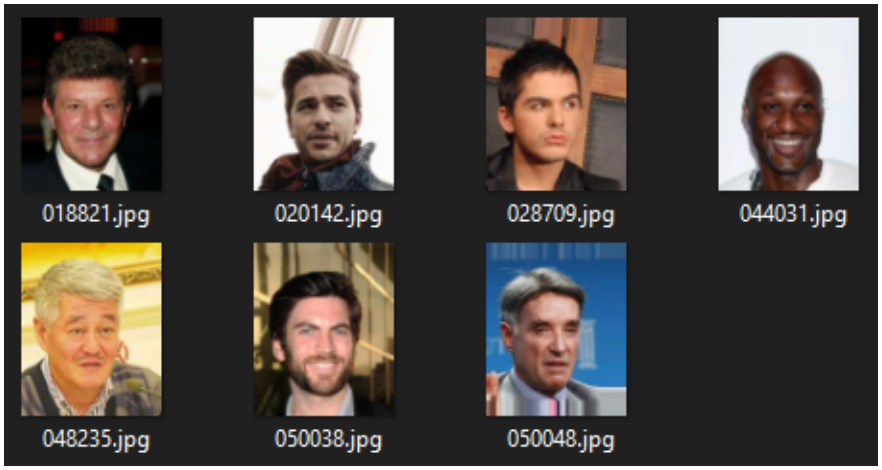
شکل ۴۱: ۲ cluster



شکل ۴۲: ۴ cluster



شکل ۴۳: cluster ۵



شکل ۴۴: cluster ۶

با توجه به توضیحات ارائه شده در فاز ۳ برای هر کلاستر، تحلیل خوشه بندی داده های تست به صورت زیر است

- **cluster ۰ :** هیت مپ کاملا یکسان است جز ویژگی bald که تنها در ۲درصد تفاوت دارند.
- **cluster ۱ :** هیت مپ نسبتا یکسان است و در درصد کمی تفاوت دارند جز ویژگی ته ریش که نسبتا ۲۳ درصد تفاوت دارند. این تفاوت میتواند به دلیل کم بودن داده های تست و کمبود تنوع در آن باشد. همچنین میتوان مشاهده کرد که در داشتن بینی تیز اشتراک دارند.

- **cluster ۲ :** دو عکس انخاب شده از داده های تست قابل مشاهده است که در ویژگی لب بزرگ اشتراک دارند. همچنین با هیتمپ بخش train فقط در۶ درصد تفاوت دارند که ممکن است از نویز باشد.
- **cluster ۳ :** در داده های تست هیچ داده ای مربوط به این خوشه نبود.
- **cluster ۴ :** هیتمپ هردو بخش شامل دو ویژگی بارز بینی نوک تیز و لب بزرگ هستند اما درصد تفاوت آنها در ویژگی های دیگر جمعا حدود ۶۰درصد است که دلیل آن میتواند دلایل ذکر شده و همچنین توزیع کمتر داده های تست باشد.
- **cluster ۵ :** این خوشه هم در داشتن ویژگی blurry هیتمپ یکسان دارند و تفاوت آنها در نداشتن مو و لب بزرگ است. داده های test که مربوط به این خوشه هستند نسبت با داده های train بیشتر دارای ویژگی مو داشتن . لب بزرگ هستند.
- **cluster ۶ :** این خوشه هم در داشتن ویژگی ته ریش اشتراک کامل دارد و بخش test کاملا ۱ و -۱ است. بخش train به دلیل داشتن داده های بیشتر تفاوت جزئی در درصد این ویژگی ها دارد مثلا در بخش train افرادی وجود داشته اند که دارای لب بزرگ هستند ولی در بخش test اینگونه نیست.

در کل با توجه به نتیجه هیتمپ میتوان نتیجه گرفت خوشه بندی به صورت خیلی خوبی انجام شده و به دلیل تنوع بیشتر در داده های train و یا وجود نویز، اختلاف بسیار کمی در درصد این ویژگی برای هیت مپ وجود دارد