مستندات پروژه هوش محاسباتی

اعضای گروه: الناز محمدی، زهرا رستمی

فاز اول:feature extraction

در فاز ابتدایی پروژه هوش محاسباتی، هدف استخراج ویژگی‌های چهره، شامل رنگ پوست و رنگ چشم، از تصاویر یک دیتاست است. ورودی ما مجموعه تصاویر dataset **است (10 عدد عکس برای نمونه انتخاب شده و در پوشه** samples **قرار دادیم) خروجی تصاویری با چهره‌های شناسایی شده که روی آن‌ها مستطیل‌هایی برای نشان دادن محل چهره کشیده شده و در پوشه** processed_images_samples **ذخیره می‌شوند. همچنین، قرار است ویژگی‌هایی مانند رنگ پوست و رنگ چشم از تصاویر استخراج شود.**

برای اجرای این کد، نیاز به نصب و راه‌اندازی برخی کتابخانه‌ها داریم:

MTCNN **: برای تشخیص چهره‌ها.**

OpenCV **: برای خواندن و پردازش تصاویر.**

Matplotlib **: برای نمایش تصاویر**

Pandas **: برای تجزیه و تحلیل داده‌ها**

ProcessPoolExecutor و ThreadPoolExecutor : برای اجرای موازی کارها در پایتون

در بخش تشخیص چهره ، با استفاده از MTCNN **چهره‌ها از تصاویر ورودی شناسایی می‌شوند و سپس یک مستطیل دور هر چهره کشیده می‌شود .در نهایت، تصاویر پردازش شده در پوشه مشخص شده ذخیره می‌شوند.**

برای اطمینان از صحت و شناسایی چهره‌ها با استفاده از یک شرط بررسی می‌کنیم که اگر تعداد چهره‌های شناسایی شده در تصویر بیش‌تر از یکی بود پیامی شامل عکس و تعداد چهره‌های شناسایی شده به ما نشان دهد.

اگر در تصویری چهره‌ای شناسایی نشد، الگوریتمی مثل Facial Landmark Detection می‌تواند نقاط کلیدی صورت مانند چشم‌ها، بینی و دهان را پیدا کند. به همین دلیل قطعه کدی برای مواقعی که اگر چهره‌ای توسط روش اول شناسایی نشود، بتوان با آن روش نقاط کلیدی صورت پیدا شود نیز پیاده‌سازی شده است.

پس از تشخیص چهره برای تشخیص رنگ پوست و چشم از تصاویر پردازش‌شده استفاده می‌شود. ابتدا، چهره‌ها شناسایی می‌شوند، سپس از ناحیه‌ی چهره برای استخراج رنگ پوست و از ناحیه‌ی چشم برای استخراج رنگ چشم استفاده می‌شود. در نهایت، این ویژگی‌ها به یک فایل CSV اضافه شده و به‌روزرسانی می‌شود. از آنجایی که حجم دیتاست ما بالاست برای افزایش سرعت پردازش، از روش مالتی تردینگ نیز استفاده شده است.

استخراج رنگ پوست: از فضای رنگی YCrCb برای تشخیص رنگ پوست استفاده می‌شود. محدوده‌ی رنگ پوست در این فضا مشخص شده است. با استفاده از ماسک رنگ پوست، پیکسل‌های مربوط به پوست شناسایی می‌شوند و رنگ میانه آن‌ها به‌عنوان رنگ پوست انتخاب می‌شود.

استخراج رنگ چشم: با استفاده از توابع find_eyes و extract_eye_colors، چشم‌ها شناسایی می‌شوند و رنگ غالب عنبیه محاسبه می‌شود.اگر رنگ غالب شناسایی نشود، یک مقدار پیش‌فرض به‌عنوان رنگ چشم ثبت می‌شود. در نهایت، مقادیر استخراج شده‌ی رنگ پوست و چشم در فایل CSV به‌روز می‌شوند.

فاز دوم: feature selection

در فاز دوم پروژه هوش محاسباتی، هدف استفاده از همبستگی بین ویژگی‌های مختلف برای انتخاب بهترین ویژگی‌ها است هدف نهایی انتخاب حداقل 6 ویژگی برتر از بین ویژگی‌های موجود برای خوشه‌بندی است. برای این کار ابتدا ویژگی‌های اولیه را با استفاده از یک ماتریس همبستگی پیرسون بررسی کرده و سپس ویژگی‌هایی که با سایر ویژگی‌ها همبستگی کمتری باهم دارند انتخاب می‌شوند.

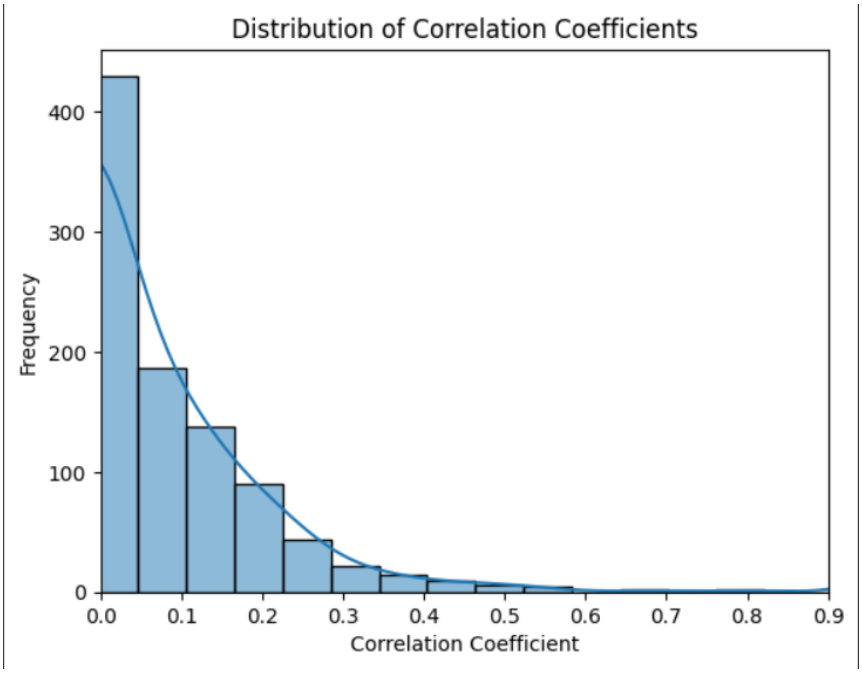
برای بررسی ارتباط بین دو ویژگی، از ضریب همبستگی پیرسون استفاده می‌شود. فرمول آن به صورت زیر است:

$$\frac{\sum(\bar{y} - y)(\bar{x} - x)}{\sqrt{\sum(\bar{y} - y)^2 \times \sum(\bar{x} - x)^2}} = xy^r$$

این فرمول ارتباط بین دو متغیر x و y را با مقایسه انحراف آن‌ها از میانگینشان اندازه‌گیری می‌کند.

با استفاده از تابع compute_correlation_matrix **، یک ماتریس همبستگی بین تمامی ویژگی‌ها محاسبه می‌شود. هر عنصر در این ماتریس، مقدار همبستگی پیرسون بین دو ویژگی را نشان می‌دهد. در صورتی که دو ویژگی همبستگی بالایی داشته باشند (مقدار نزدیک به 1 یا –1)، این بدین معناست که این دو ویژگی به یکدیگر وابسته هستند.**

با کمک نمودار توزیع ضرایب همبستگی ، یک آستانه برای انتخاب بهترین ویژگی انتخاب می‌کنیم که با توجه به نمودار ما عددی بین 0.1 **تا** 0.2 **می‌تواند انتخاب مناسبی باشد.**



با توجه به نمودار ما اکثر ضرایب همبستگی نزدیک به صفر هستند و با افزایش مقدار ضریب همبستگی، تعداد آنها به‌طور نمایی کاهش می‌یابد. به عبارت دیگر، تعداد بالایی از جفت داده‌ها همبستگی ضعیفی با هم دارند و تعداد کمی دارای همبستگی قوی هستند.

ویژگی‌هایی که همبستگی بالا دارند حذف می‌شوند.

فاز سوم: Clustering

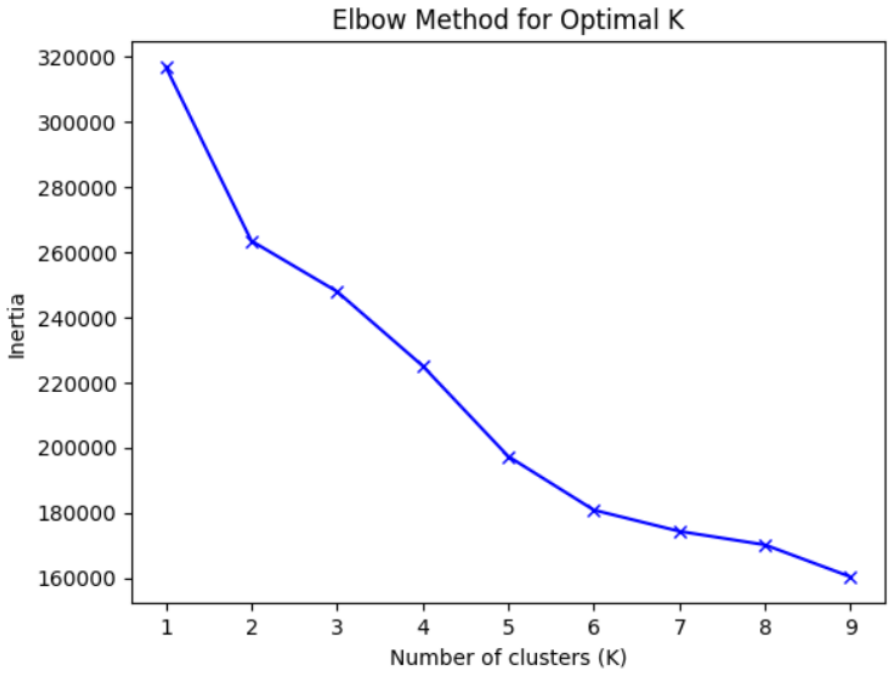
در این فاز با استفاده از سه الگوریتم KMeans، DBSCAN **و** MeanShift **، داده‌های خود را خوشه‌بندی می‌کنیم. هدف این است که با انجام** hyperparameter tuning **برای هر الگوریتم، بهترین مقادیر برای خوشه‌بندی را پیدا کنیم. در نهایت، با استفاده از معیارهای ارزیابی (مانند** **Silhouette Score**) **کیفیت هر الگوریتم را بررسی و مقایسه می‌کنیم.**

همچنین در این فاز از **Heatmap** **برای مشاهده ویژگی‌های بارز در هر خوشه استفاده شده است.**

خوشه‌بندی با KMeans :

با استفاده از الگوریتم KMeans **و تنظیم پارامتر** n_clusters= 3**، داده‌ها خوشه‌بندی شده‌اند.(این پارامتر تعداد خوشه‌ها را مشخص می‌کند)**

برای تعیین تعداد بهینه خوشه‌ها (K) **در الگوریتم** KMeans **، از روش** **Elbow Method** **استفاده شده است. در این روش، ابتدا الگوریتم** KMeans **را با مقادیر مختلف** K **اجرا می‌کنیم و میزان** Inertia **را برای هر مقدار** K **محاسبه می‌کنیم.** Inertia **معیاری است که نشان می‌دهد داده‌ها چقدر به مراکز خوشه‌های خود نزدیک هستند. نقطه‌ای که میزان** Inertia **به‌طور ناگهانی کاهش می‌یابد به عنوان** k **انتخاب می‌شود.**



معیار ارزیابی Silhouette Score برای ارزیابی کیفیت خوشه‌بندی محاسبه شده است.

این مقدار بین –1 و 1 متغیر است:

نزدیک به 1: به معنای این است که نمونه به خوشه مناسبی تعلق دارد و خوشه‌بندی به خوبی انجام شده است.

نزدیک به 0: نمونه‌ها در مرز خوشه‌ها قرار دارند و تمایز بین خوشه‌ها مشخص نیست.

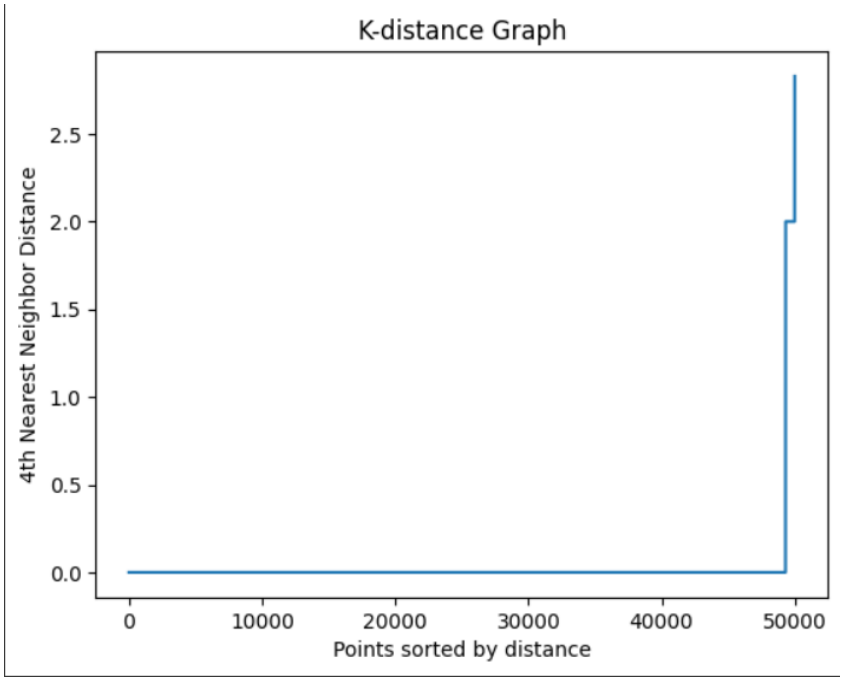
نزدیک به 1–: نمونه‌ها در خوشه نادرستی قرار دارند.

برای ارزیابی کیفیت کلی خوشه‌بندی، میانگین Silhouette Score تمام داده‌ها محاسبه می‌شود. هرچه این میانگین بیشتر باشد، خوشه‌بندی بهتری رخ داده است.

هرچه رنگ‌ها در Heatmap تیره‌تر باشند، مقدار ویژگی بالاتر است. این نمودار تفاوت‌های بین خوشه‌ها را به راحتی نشان می‌دهد. اگر یک ویژگی در یک خوشه مقدار بیشتری داشته باشد، می‌توان نتیجه گرفت که این ویژگی نقش مهمی در شکل‌گیری آن خوشه داشته است.

خوشه‌بندی باDBSCAN :

الگوریتم DBSCAN را با پارامترهای eps=1.9 و min_samples=5 اجرا می‌کنیم. این پارامترها تعیین می‌کنند که چه نقاطی به عنوان نویز در نظرگرفته شوند و چگونه خوشه‌ها شناسایی شوند. مجدداً، معیار Silhouette Score برای ارزیابی این الگوریتم استفاده شده است. برای تنظیم بهتر پارامتر eps درDBSCAN ، از نمودار K-Distance استفاده شده است. این نمودار کمک می‌کند مقدار مناسبی برای eps انتخاب کنیم.



نحوه کارکرد آن به این صورت است که ابتدا برای هر نقطه در داده‌ها، فاصله آن تا k نزدیک‌ترین همسایه آن محاسبه می‌شود. سپس فواصل محاسبه‌شده به ترتیب صعودی مرتب می‌شوند. پس از آن این فواصل بر روی نموداری با محور x که نقاط مرتب‌شده را نشان می‌دهد و محور y که فاصله‌ها را نشان می‌دهد رسم می‌شود. نقطه‌ای که نمودار به شکل ناگهانی افزایش پیدا می‌کند نقطه‌ای است که مقدار بهینه eps را نشان می‌دهد. این نقطه نشان می‌دهد که از آن به بعد نقاط در فاصله‌ای دورتر قرار دارند و بنابراین باید خوشه‌ها از یکدیگر جدا شوند.

نکته‌ای که وجود دارد این است که استفاده از الگوریتم dbscan برای داده های باینری، انتخاب خوبی نمی‌باشد به همین دلیل در نتیجه خوشه بندی ما نیز تعداد خوشه‌های زیاد با نویز بالا را شاهد هستیم.

خوشه‌بندی باMeanShift :

الگوریتم MeanShift با تنظیم پارامتر bandwidth = 2.8 اجرا شده است. مانند دو الگوریتم قبلی، Silhouette Score برای ارزیابی کیفیت خوشه‌بندی محاسبه شده است.

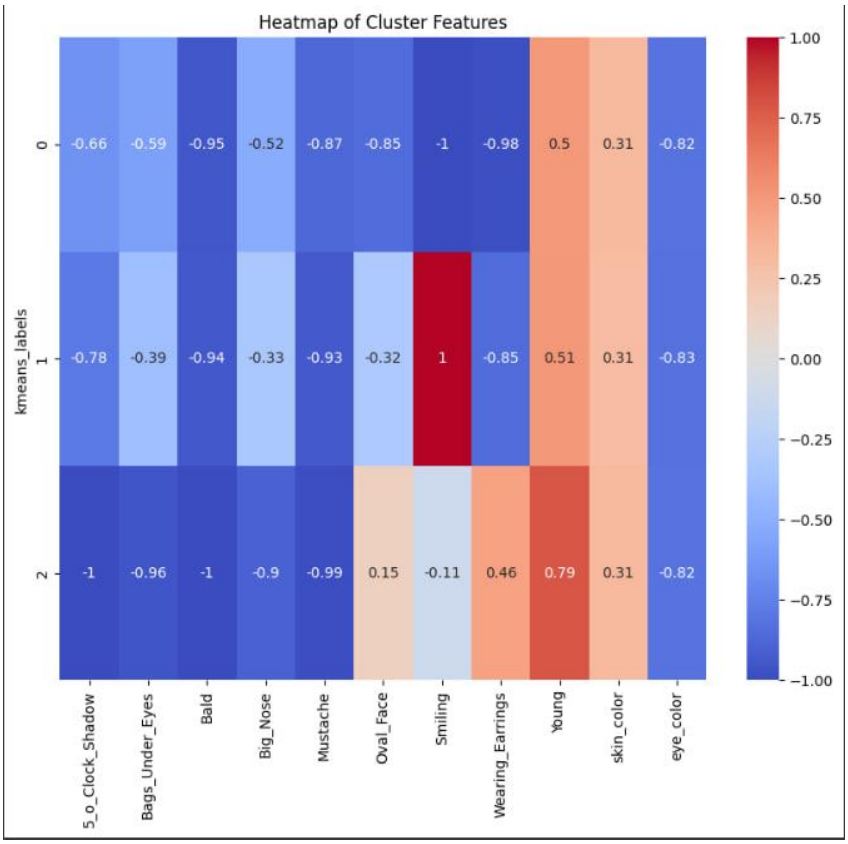
برای هر سه الگوریتم Silhouette Score را محاسبه کردیم و از آنجایی که بالاترین نمره نشان‌دهنده‌ی الگوریتم بهتر برای ساختار خوشه‌ای این دیتاست هست، می‌بینیم نتیجه

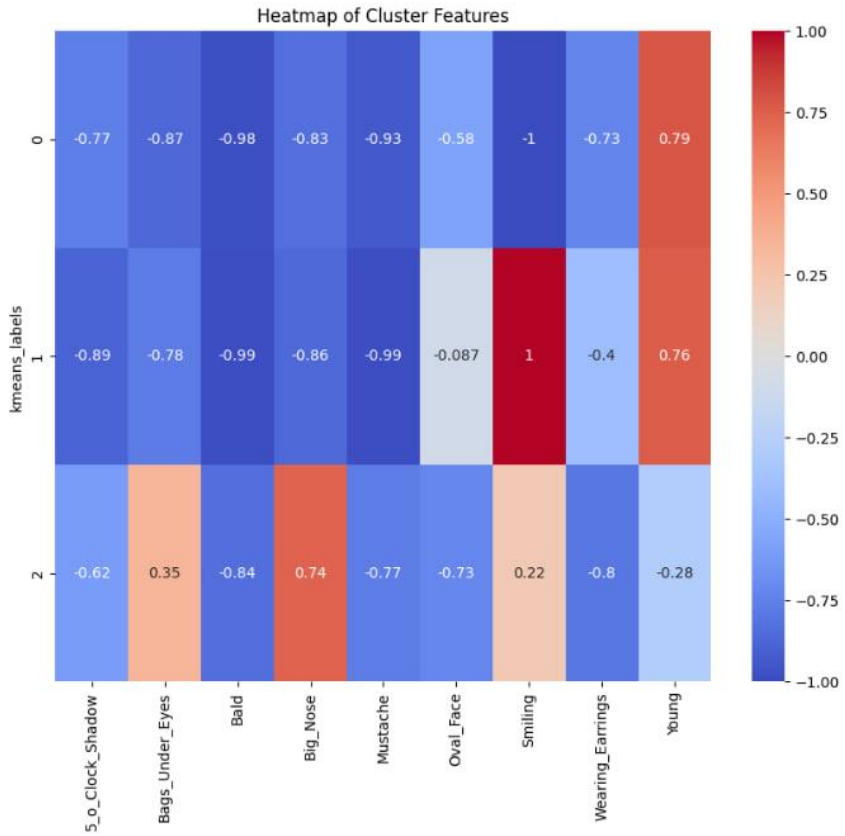
الگوریتم‌های meanshift , kmeans تقریبا مشابه و کمتر از dbscan است.

باید توجه داشت که الگوریتم kmeans برای دیتاست‌های با نویز یا داده‌های با تراکم متغیر عملکرد خوبی ندارد و به عکس الگوریتم dbscan برای داده‌هایی با تراکم‌های متفاوت و نویز زیاد مناسب‌تر است.

از آنجایی که برای کاهش تعداد کلاسترهای خروجی در dbscan **(نمایش راحت تر) تعداد** min-sample **ها را زیاد در نظر گرفتیم داده‌های نویز بیش‌تری را در نتیجه این الگوریتم داریم و از لحاظ منطقی Silhouette Score ما نیز برای این الگوریتم نمره بیش‌تری را می‌گیرد.**

الگوریتم k-means **را یک بار با انتخاب ستون های رنگ پوست و رنگ چشم و یک بار بدون آنها اجرا می‌کنیم. در زیر** heatmap **های مربوط به هر یک از اجراهارا می‌بینیم:**





طبق نتیجه خروجی می‌بینیم در حالتی که از رنگ پوست و چشم استفاده کردیم، این دو عامل اثیر مشخصی بر برخی از خوشه‌ها گذاشته‌اند و می‌توانند بر جداسازی خوشه‌ها تأثیر داشته باشند اما درحالتی که این دو ویژگی را درنظر نگرفتیم ، بقیه ویژگی‌ها برای خوشه‌بندی اهمیت بیش‌تری پیدا کرده‌اند و وزن این ستون‌ها کاهش پیدا کرده است. همانطور که می‌بینیم در حالتی که رنگ پوست و رنگ چشم را درنظر نگرفتیم، منجر به تغییرات کوچکی در توزیع ویژگی‌ها شده است. این تغییرات می‌تواند نشان‌دهنده‌ی تأثیر کم این دو ستون بر خوشه‌بندی باشد، که با حذفشان، ویژگی‌های دیگر فرصت بیشتری برای تاثیرگذاری پیدا کرده‌اند. از آنجایی که ویژگی‌های دیگری وجود دارند که مقداری نزدیک به 1 و 1- دارند، تاثیر بیش‌تری درخوشه‌بندی دارند. یعنی مدل ما با در نظرنگرفتن رنگ چشم و رنگ پوست بازهم الگوهای قوی‌ای برای خوشه‌بندی دارد.

به طور کلی می‌توان گفت رنگ چشم و رنگ پوست تاثیر کلی کمی روی نتایج خوشه‌بندی ما دارد.

در بخش دیگر برای هر سه الگوریتم اجرایی 10 عکس از هر خوشه را جدا کرده و در فولدر مشخصی ذخیره می‌کنیم. طبق heatmapهایی که برای الگوریتم داشتیم و نتایج 10 عکس از هر خوشه می‌توان تحلیل زیر را داشت:

الگوریتم k_means: در این الگوریتم ما 3 خوشه داریم که با نگاه به heatmap آن در خوشه 0 برای مثال می‌بینیم ویژگی لبخند زدن مقدار 1- دارد و نتیجه می‌گیریم عکس‌هایی که در فولدر مربوط به این خوشه ذخیره شده اند باید حداقل ویژگی متمایزکننده عدم لبخند زدن را داشته باشد با بررسی تصاویر ذخیره شده می‌بینیم در خروجی، این ویژگی درمورد عکس ها صادق است. به همین صورت برای خوشه 1 می‌بینیم ویژگی لبخند زدن مقدار یک را دارد یعنی انتظار می‌رود تصاویر ذخیره شده در حال لبخند زدن باشند که با بررسی آن‌ها می‌بینیم به همین صورت است. به همین روال برای خوشه 2 می‌بینیم ویژگی‌های بدون مو بودن و سیبیل داشتن عددی نزدیک به 1- دارند و ویژگی جوان بودن عددی نزدیک به 1 پس توقع می‌روند در تصاویر این خوشه، خانم هایی که سن جوان تا میانسال دارند را شامل شود و با بررسی تصاویر صحت این موضوع مشخص می‌شود.

الگوریتم dbscan : مانند توضیحات قبلی در این الگوریتم هم باتوجه به heatmap و 25 خوشه ایجاد شده تصاویر در 25 فولدر قرار گرفته. (به دلیل تعداد زیاد به بررسی و مثال زدن درمورد یک خوشه و فولدر می‌پردازیم) مثلا در خوشه 4 مقدار داشتن گوشواره و جوان بودن 1 است یعنی تصاویر باید این ویژگی را داشته باشند و به همین منوال مقادیر لبخند زدن و بی‌مو بودن 1- است یعنی نباید این ویژگی‌ها را داشته باشند پس توقع داریم تصاویر خانم‌های جوانی که گوشواره دارند و لبخند نمی‌زنند در این فولدر قرار داشته باشند که با بررسی تصاویر می‌بینیم این ویژگی‌ها رعایت شده است.

الگوریتم meanshift : مانند توضیحات قبلی در این الگوریتم هم باتوجه به heatmap و 2 خوشه ایجاد شده تصاویر در 2 فولدر قرار می‌گیرند. مثلا در خوشه 0 ، ویژگی جوان بودن عددی نزدیک به 1 دارد و به عکس ویژگی بی‌مو بودن و بینی بزرگ عددی ندیک به 1- دارند پس توقع می‌رود در تصاویر فولدر مربوط به خوشه 0، تصاویر دارای این ویژگی‌ها باشند که با بررسی تصاویر می‌بینیم این اتفاق افتاده است. در خوشه یک ویژگی بینی بزرگ داشتن عددی نزدیک به 1 است و ویژگی بی‌مو بودن عددی نزدیک به 1- دارد پس انتظار می‌رود که تصاویری که در فولدر مربوط به خوشه 1 قرار گرفته اند بینی تقریبا بزرگ داشته و کچل نباشند.

فاز چهارم: Visualization

کاهش ابعاد با استفاده ازPCA :

یک تکنیک خطی برای کاهش ابعاد است که داده‌ها را به فضایی با ابعاد کمتر تبدیل می‌کند، به طوری که بیشترین واریانس داده‌ها حفظ شود. داده‌ها به دو مؤلفه اصلی تبدیل می‌شوند. مؤلفه اول بیشترین واریانس داده‌ها را می‌گیرد و هر مؤلفه بعدی باقی‌مانده واریانس را می‌گیرد.

پس از کاهش ابعاد داده‌ها به ۲ مؤلفه با استفاده ازPCA ، الگوریتم KMeans برای خوشه‌بندی داده‌ها اعمال می‌شود. پس از خوشه‌بندی، خوشه‌ها در فضای دوبعدی ایجادشده توسط PCA رسم می‌شوند. نقاط داده با استفاده از sns.scatterplot رسم شده و هر نقطه بر اساس خوشه‌ای که به آن اختصاص داده شده، رنگ‌آمیزی می‌شود.

هر چه درصد واریانس PCA بیشتر باشد، نشان می‌دهد که مؤلفه‌های انتخابی اطلاعات بیشتری از داده‌ها را در خود جای داده‌اند. با درصد پایینی که در خروجی نتایج مشاهده می‌کنیم، احتمالاً ویژگی‌های دیگر اطلاعات زیادی دارند که در ابعاد کاهش‌یافته از دست رفته است.

فاز پنجم: K-Means وKNN

ابتدا برای 50 داده نزدیک مرکز کلاستر، نتیجه اجرا KNN نشان می‌دهد که تمامی داده‌ها در کلاسترهای درست قرار گرفته اند.
این کار را برای 3000 داده نزدیک مرکز کلاستر نیز تکرار می‌کنیم و نتیجه مجددا مشابه دفعه قبلی می‌باشد و مشخص می‌کند که تمامی داده های در خوشه‌های مناسب و درستی قرار گرفته‌اند.

از آنجایی که در هر دو تست، نتایج مشابهی به دست می‌آید، نشان می‌دهد که خوشه‌های ایجادشده پایدار و قابل اعتماد هستند. یعنی داده‌ها به خوبی حول مرکز هر خوشه متمرکز شده‌اند و پراکندگی کمی نسبت به مرکز خوشه دارند. نتیجه نشان می‌دهد که تعداد خوشه‌ها (K) نیز به‌درستی انتخاب شده و خوشه‌های متمرکزی ایجاد شده‌اند که نمایانگر ساختار واقعی داده‌ها هستند.

فاز ششم: Prediction

در این فاز قصد داریم با توجه به دیتاست تست، تصاویر را بررسی می‌کنیم که مربوط به کدام خوشه است. برای هر 10 داده ، نتایج را ویژال می‌کنیم در کنار آن 5 داده مشابه با آن را نیز ویژال می‌کنیم. در نهایت یک ستون برای خوشه‌ای که مربوط به تصویر هست اضافه می‌کنیم.
با مقایسه heatmap فاز 3 که مربوط به الگوریتم k-means هست برای خوشه 0 مثلا باید تصاویری در آن قرار بگیرند که عینکی نباشند و بدون مو نباشند که با مشاهده نمونه‌هایی کع ویژال شده‌اند می‌بینیم این ویژگی رعایت شده است.
برای خوشه 1 ، افراد با پوست روشن ، بینی کوچک و ابروهای کم‌پشت باید باشند که با مشاهده نمونه‌های ویژوال شده می‌بینیم این ویژگی‌ها با نسبت خوبی در نمونه‌های ما مشاهده می‌شوند.

برای خوشه 2 ، می‌توان گفت افرادی با بینی‌های به نسبت بزرگ، ابروهای تقریبا کم‌پشت هستند. البته باید توجه داشت خوشه دوم ویژگی‌های متمایز کننده شدیدی ندارد به همین دلیل تنوع مقادیر و نمونه‌ها در آن بیش‌تر خواهد بود که این مورد در نمونه‌های خروجی نیز مشاهده می‌شود.