به نام خدا

مهرانه مقتدائی فر ۹۷۲۲۲۰۸۶

تاریخ: ۱۴۰۰/۹/۲۰

گزارش تمرین سری ۳ **درس: مبانی علوم داده** **استاد درس: دکتر خردپیشه**

در این تمرین از ما خواسته شده که یک Music recommender پیاده کنیم. اینکار را با استفاده از داده هایی برای train کردن یک مدل clustering شروع میکنیم.

در این دیتاست، داده هایی از audio feature های 42 هزار آهنگ مختلف از spotify جمع آوری شده است. این آهنگ ها ویژگی های متفاوتی دارند و همچنین genre آنها نیز مشخص است. ما این داده ها را در نظر میگیریم و با استفاده از آنها یک مدل خوشه بندی مناسب را ایجاد میکنیم که با توجه به نزدیکی این داده ها در فضا، بتواند آنها را در دسته‌بندی های مختلف قرار دهد.

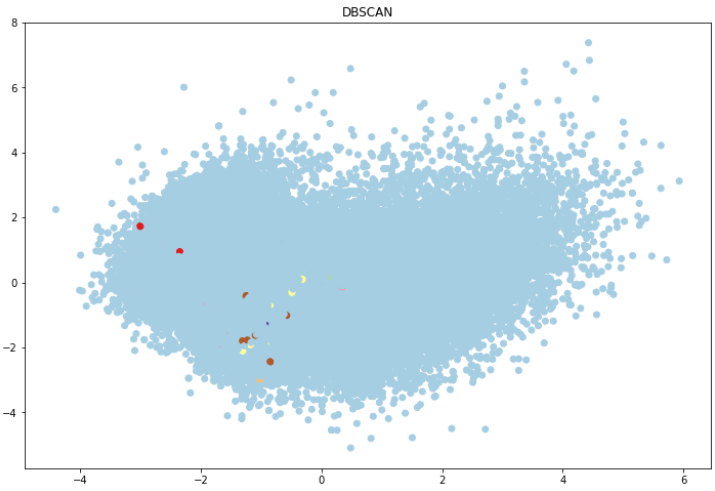
در ویژگی هایی که از دیتاست جدا کردیم، یک ستون به نام genre وجود دارد که تنها ستون categorical ما در این دیتاست می‌باشد، این ویژگی را یک بار با استفاده از LabelEncoding و بار دیگر با استفاده از onehot encoding به داده numeric تبدیل کردیم.

در ابتدا ویژگی genre را با استفاده از labelencoding، تبدیل به ۱۵ کتگوری مختلف از ۰ تا ۱۴ کردیم. سپس برای آنکه تمامی داده ها در یک رنج مشخص قرار بگیرند آنها را با استفاده از StandardScaler، scale کردیم.

پس از اینکار، ۳ مدل مختلف را برای خوشه بندی بر روی این داده ها پیاده سازی کردیم و برای نمایش آنها، با استفاده از PCA کاهش بعد انجام داده و در ۲بعد نمایش دادیم.

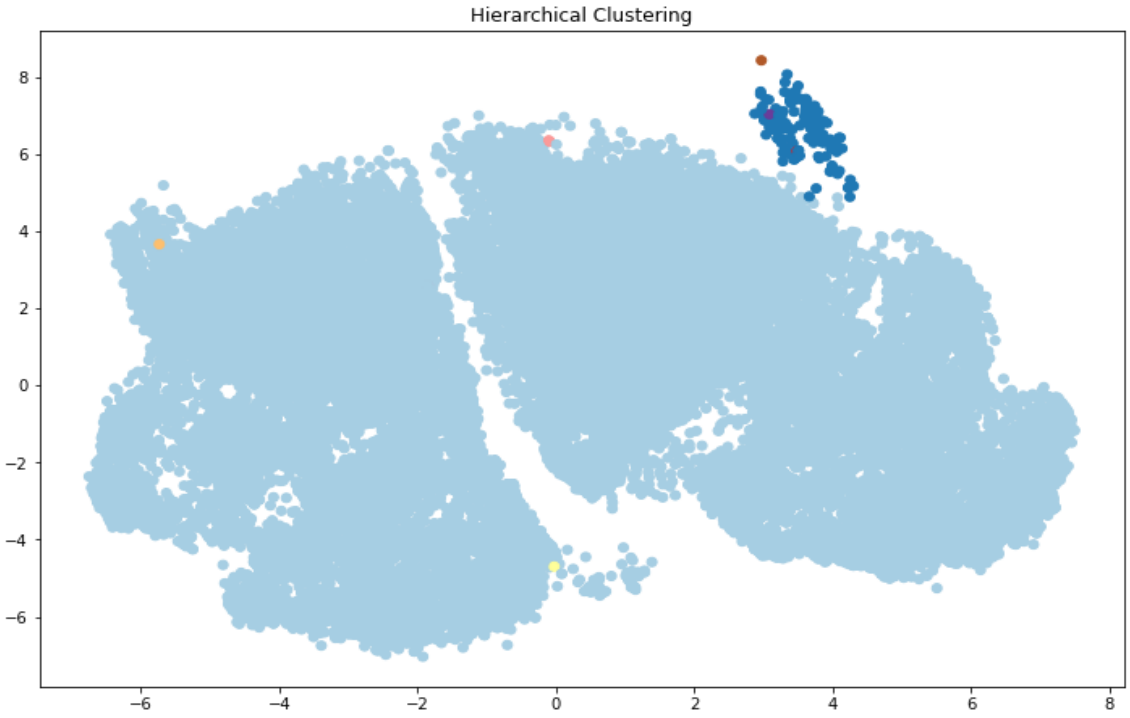
برای الگوریتم DBSCAN باید پارامتر های شعاع هر داده و همچنین مینیمم تعداد داده ها در هر خوشه را مشخص کنیم. متاسفانه با انتخاب مقادیر مختلف برای این پارامتر ها، نتیجه نهایی خیلی جالب نبود و با امتحان کردن مقادیر مختلف باز هم چون اکثر داده ها به عنوان نویز شناخته میشدند، کلاستر بندی خوبی نداشتیم.

برای مثال با انتخاب EPS = 0.003 و min_samples = 5 نمودار زیر برای خوشه ها حاصل شد:



در این صورت، داده ها نیز به ۸۱ دسته مختلف تقسیم بندی شده بودند، در حالت های دیگر حتی این تعداد دسته بندی به ۲۰۰ یا بیشتر نیز رسیده بود که اصلا نتیجه خوبی نداشت.

در ادامه الگوریتم hierarchical clustering را استفاده کردیم، این الگوریتم نیز کلاستر بندی خوبی نداشت، تعداد کلاسترهایی که در این الگوریتم مشخص کردیم، به اندازه تعداد کلاستر های بهینه در الگوریتم KMEANS بود که در ادامه آن را کاملا توضیح میدهیم.



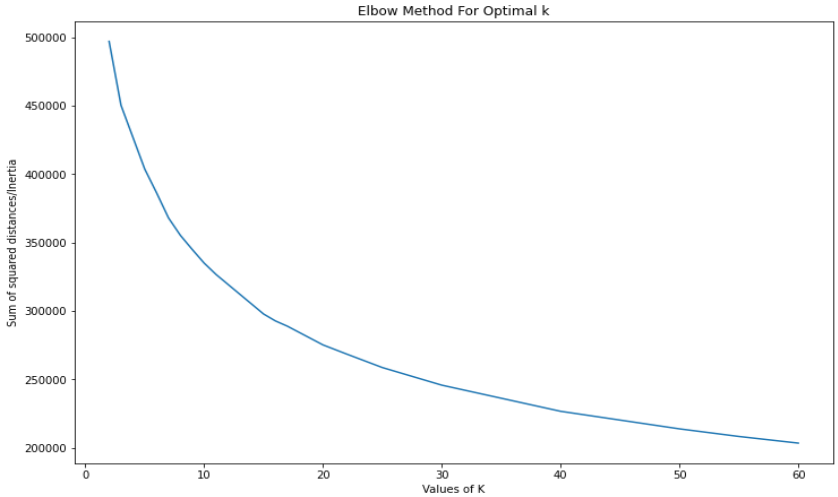
همانطور که مشاهده میکنید، این الگوریتم نیز مانند DBSCAN نتیجه مطلوبی نداشت، نوع اتصال را نیز single قرار دادیم، به این معنا که معیار سنجیدن نزدیکی کلاستر ها، مینیمم فاصله هر داده در یک کلاستر از داده‌ای در کلاستر دیگر است. اگر این Linkage را چیز دیگری قرار میدادیم، متاسفانه به علت زیاد بودن داده ها باعث crash کردن میشد.

از این به بعد تنها از الگوریتم Kmeans برای ادامه کار و یافتن نتیجه های مطلوب تر استفاده کردیم. همانطور که میدانیم در الگوریتم kmeans نیاز است تا تعداد cluster ها را مشخص کنیم. برای اینکه متوجه شویم که چه تعداد مناسب است، از دو معیار مختلف استفاده کردیم.

اولین معیار بررسی silhouette_score است، این معیار با بررسی فاصله درون خوشه‌ای و بین خوشه‌ای، میانگین آنها را مشخص میکند و هرچه این معیار بیشتر باشد یعنی فاصله کلاستر ها از هم دورتر و فاصله داده های داخل هر کلاستر بهم نزدیک تر است. برای تعداد کلاستر ها مختلف از ۲ ۴ ۵ ... الی ۲۰ ۳۰ .. ۴۰، این معیار را محاسبه کرده‌ایم. هرچه این معیار بیشتر باشد آن تعداد کلاستر را برای الکوریتم Kmeans انتخاب میکنیم.

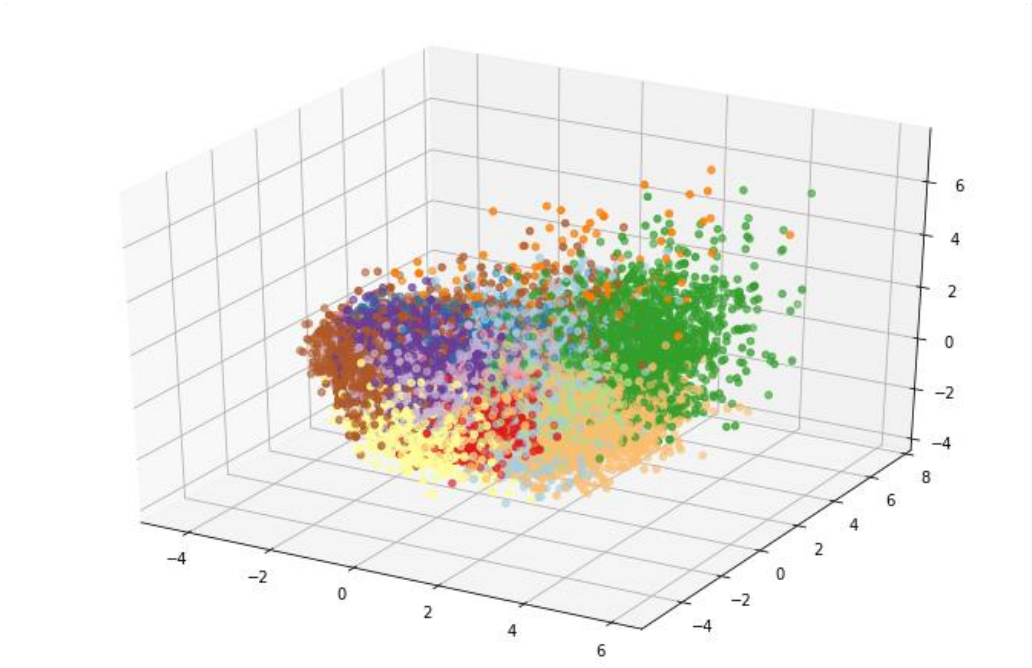
معیار دیگری که بیشتر مورد استفاده قرار گرفته است، استفاده از sum of square هست که باز هم معیار همان بررسی فاصله بین کلاستر های مختلف است، در این حالت اصطلاحا روشی به نام elbow وجود دارد که با رسم نمودار sum of square ها، در جایی که نمودار شکسته میشود یا به اصطلاح در elbow نمودار، آن تعداد کلاستر برای الگوریتم مناسب است و از آن به بعد دیگر زیاد کردن کلاستر ها تاثیری نخواهد داشت.

برای حالتی که داده ژانر را label کردیم و سپس داده ها را scale نمودیم، نمودار elbow زیر حاصل شد:

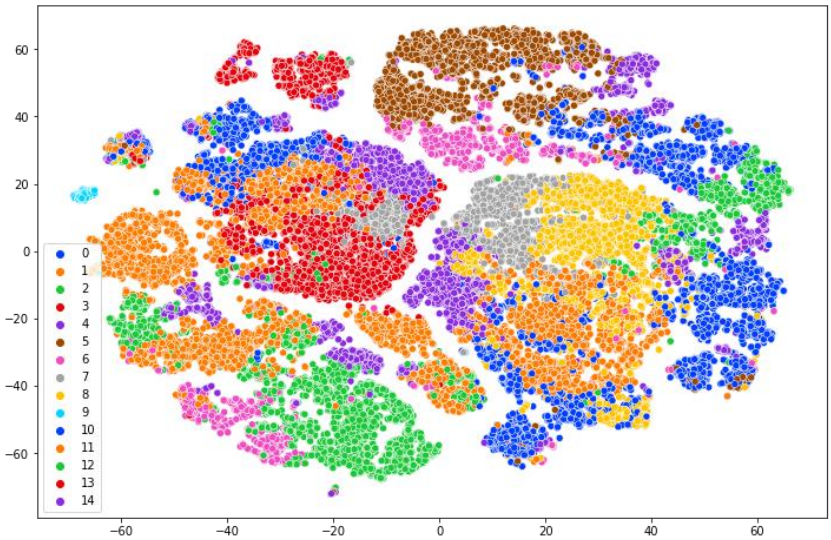


با توجه به این نمودار، تقریبا شکست آن از ۱۵ به بعد اتفاق میوفتد و ما بر فرض، تعداد کلاستر ها را ۱۵ تا درنظر میگیریم که این تعداد به اتفاق با تعداد ژانر های لیبل شده ما برابر است.

پس از آن توزیع کلاستر ها در فضا را با یکدیگر مشاهده میکنیم:

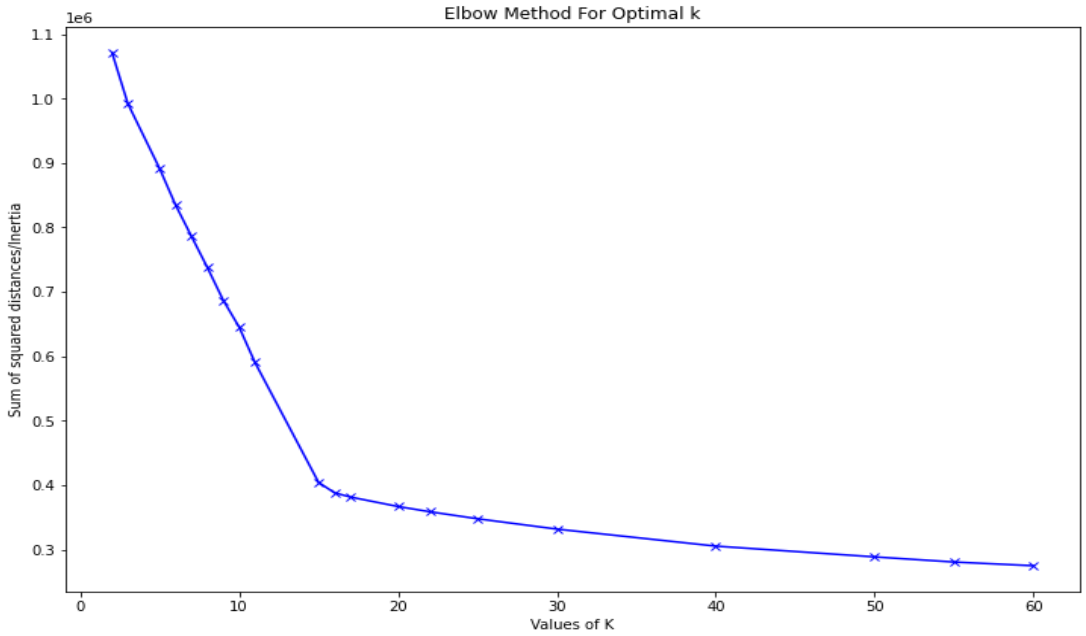


این نمودار با استفاده از PCA و کاهش به ۳ بعد رسم شده است، برای آنکه واضح تر نمایش داده شود از tSNE transform نیز برای نمایش کلاستر ها در ۲ بعد استفاده کردیم:

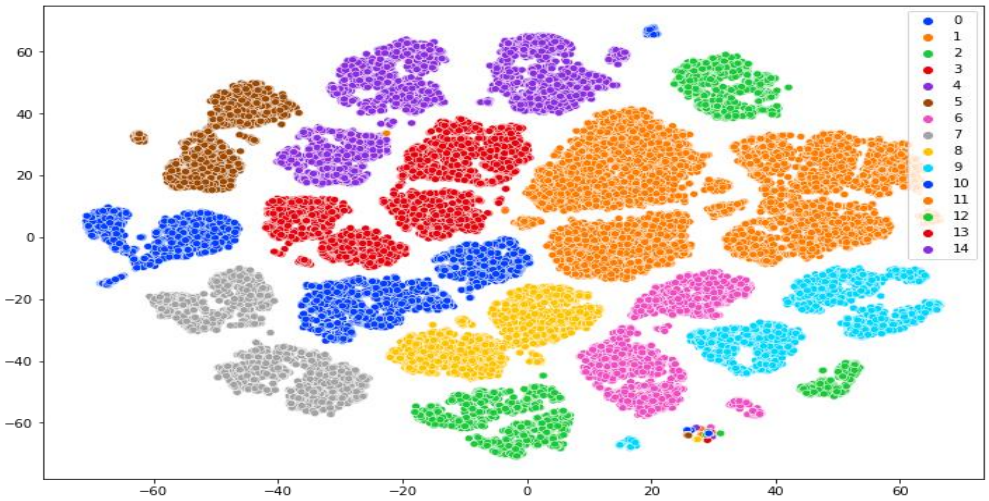


همانطور که در این شکل به طور واضح میبینیم، این نحوه خوشه بندی به درستی انجام نشده است زیرا داده ها به شکل خوبی در یک خوشه قرار نگرفته‌اند بلکه هر خوشه شامل لیبل های مختلفی از الگوریتم میباشد که این درست نیست و نشان دهنده آن است که تفکیک و یا خوشه بندی ما در این مدل خوب نمیباشد.

در مرحله بعد این دفعه ژانر ها را با استفاده از One hot encoding، به داده عددی تبدیل کردیم. با استفاده از scaling نمودار زیر روش elbow را برای این حالت نشان میدهد و به طور واضح، میبینیم که شکستگی نمودار در k=15 رخ داده است.



با استفاده از همین مسئله، تتعداد کلاستر های الگوریتم Kmeans را ۱۵ در نظر گرفته و سپس مشاهده نحوه خوشه بندی را مشاهده میکنیم:

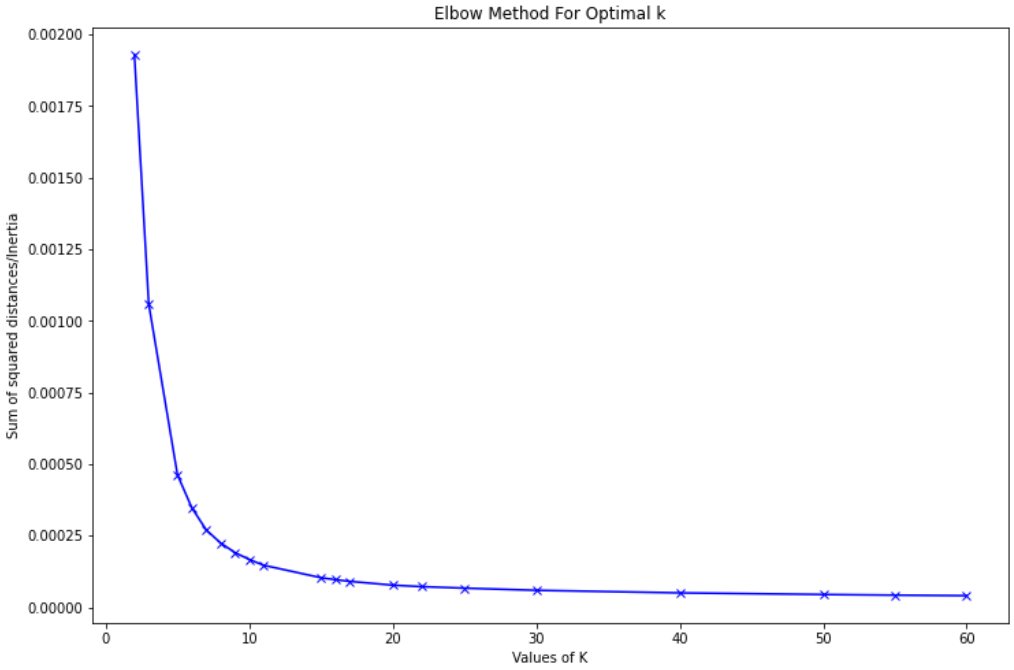


همانطور که میبینیم این نمودار از حالت قبلی بهتر شده است، در اصل انگار کلاستر ها درست تر هستند زیرا داده هایی که در یک خوشه وجود دارند تقریبا در خوشه دیگری قاتی نشده‌اند.

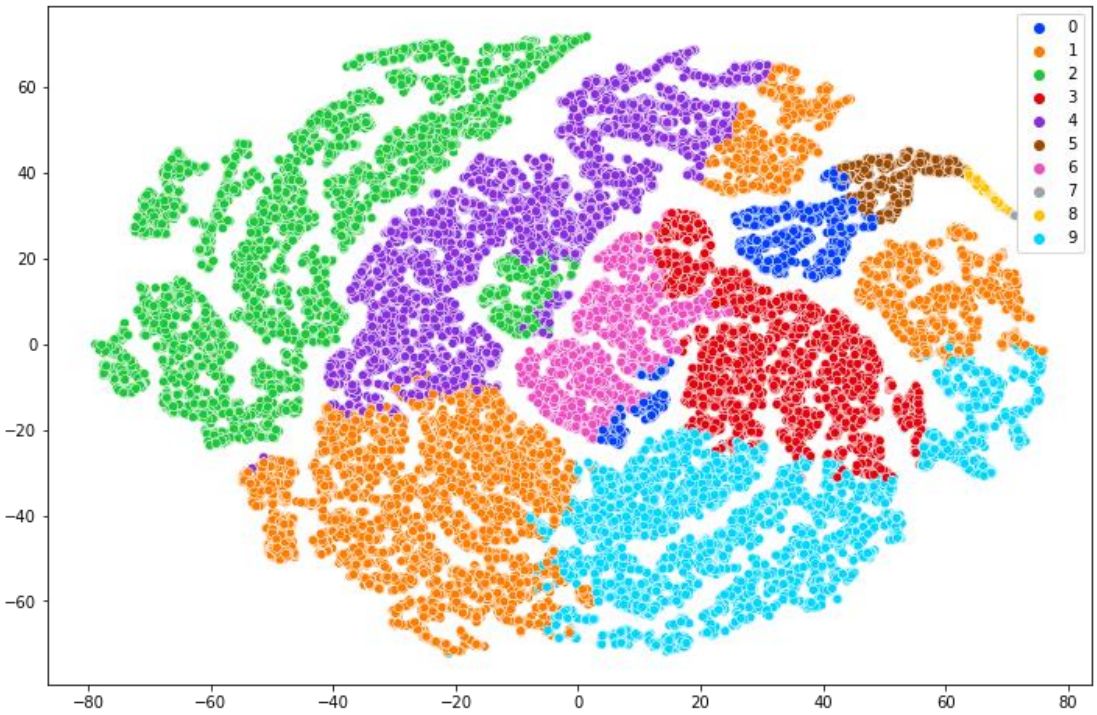
اما نکته قابل توجه آن است که در این حالت، قطعا تاثیر one hot encoding بر روی داده های ما بسیار زیاد خواهد بود و به نوعی میتوان گفت که انگار این داده ها در این حالت بر اساس ژانر جدا شده‌اند که شاید بسیار دقیق نباشد چون ویژگی های مهم دیگری نیز برای تقسیم بندی وجود دارد.

در حالت بعدی به سراغ نرمال کردن داده ها رفته‌ایم. در این حالت توزیع داده ها را به توزیع نرمال نزدیک تر کردیم. و باز هم یکبار با label encoding و با one hot امتحان کردیم.

اگر ویژگی genre را label کنیم و سپس Normalize را انجام دهیم، در این صورت خواهیم داشت:

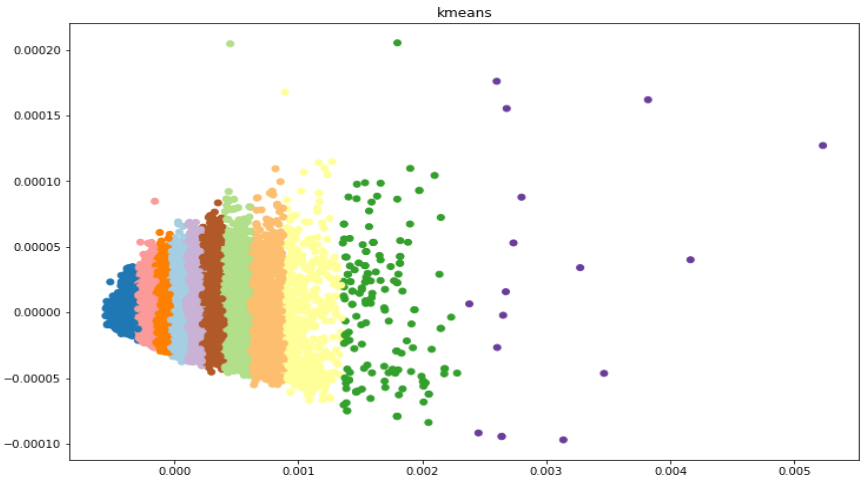


و به همین علت کلاستر بندی را با k=10 انجام میدهیم و پس از آن توزیع کلاستر ها را در فضای دو بعدی با یکدیگر مشاهده میکنیم:

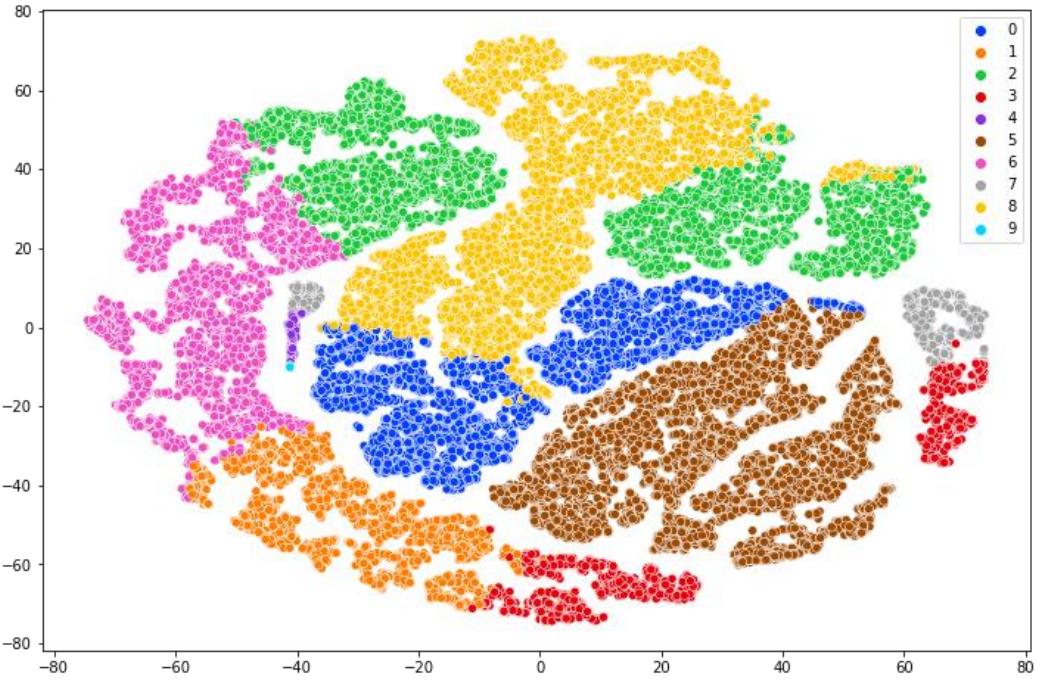


با توجه به این توزیع مشاهده میکنیم که قرار گرفتن داده ها در کلاستر های مختلف به صورت معقول میباشد و تقریبا کلاستر ها از یکدیگر جدا شده‌اند.

حال اگر داده ژانر را One hot کرده و سپس normalize را بر روی داده ها انجام دهیم، با استفاده از PCA و کاهش به ۲ بعد برای نمایش، نتیجه زیر رخ خواهد داد:



و با استفاده از TSNE نیز نتایج زیر بدست آمد:



بنظر کلاسترینگ بدی به نظر نمی‌آید اما از آنجایی که One hot کردن باعث افزایش بعد خواهد شد، بهتر است که انجام ندهیم.

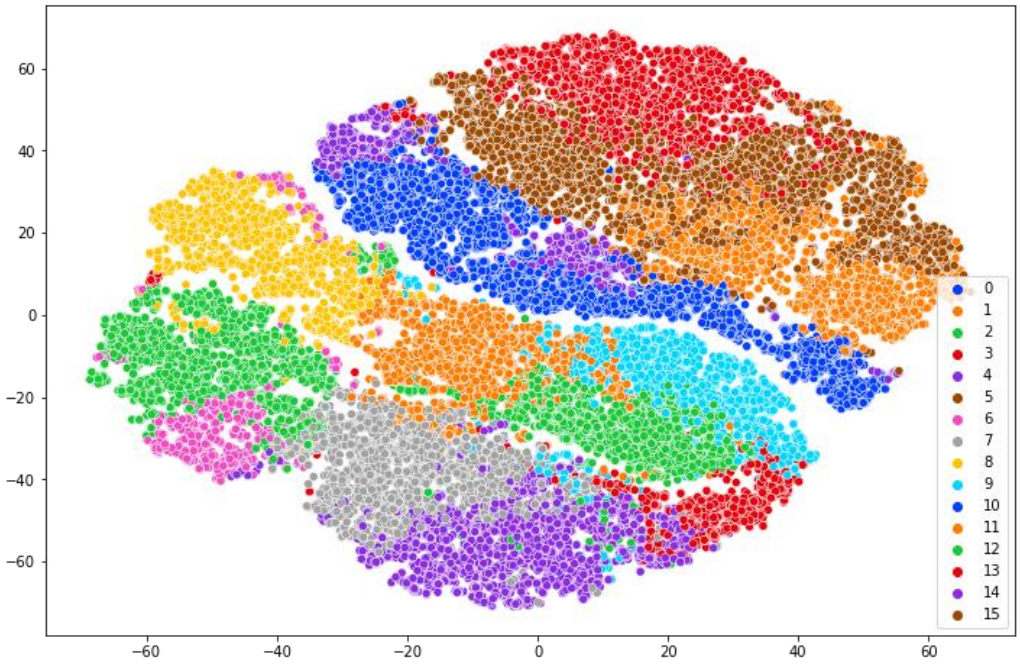
Recommender System

در انتها نیز بدون در نظر گرفتن ژانر هم این مدل را میسازیم. در نتایجی که بدست آمده مشاهده میکنیم که ژانر خیلی تاثیر زیادی بر روی خوشه بندی ندارد و در اصل با استفاده از ویژگی های دیگر نیز میتوان همان خوشه بندی ها را داشته باشیم. در ادامه نیز recommend کردن به یوزر را نیز بدون در نظر ژانر انجام خواهیم داد.

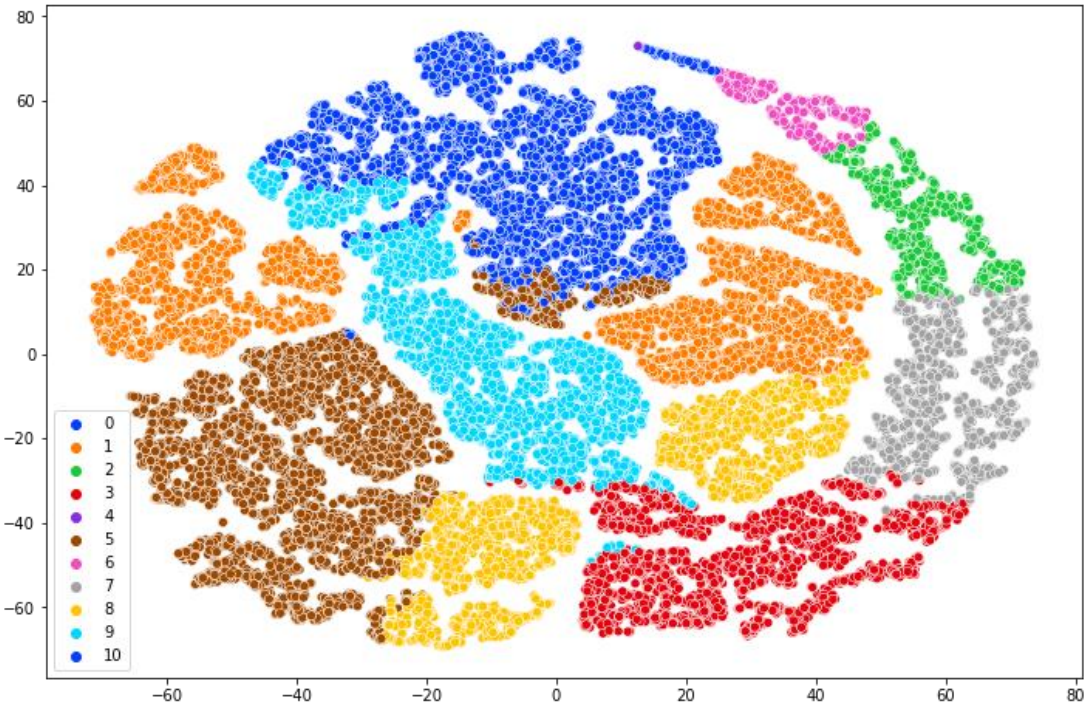
از آنجایی که داده های ورودی یک پلی لیست از آهنگ های یوزر است، نیاز هست تا با وصل شدن به API اسپاتیفای، ویژگی های این پلی لیست را بدست آوریم و داده های آن را مانند داده هایی که با آن کلاسترینگ انجام دادیم، scale یا نرمال کنیم.

در شکل های زیر به ترتیب مشاهده میکنیم که بدون در نظر گرفتن ژانر و با پیاده کردن Kmeans بر روی دیتاهایی که به صورت مختلف scale شده‌اند، کلاسترینگ به چه صورت درآمده است:

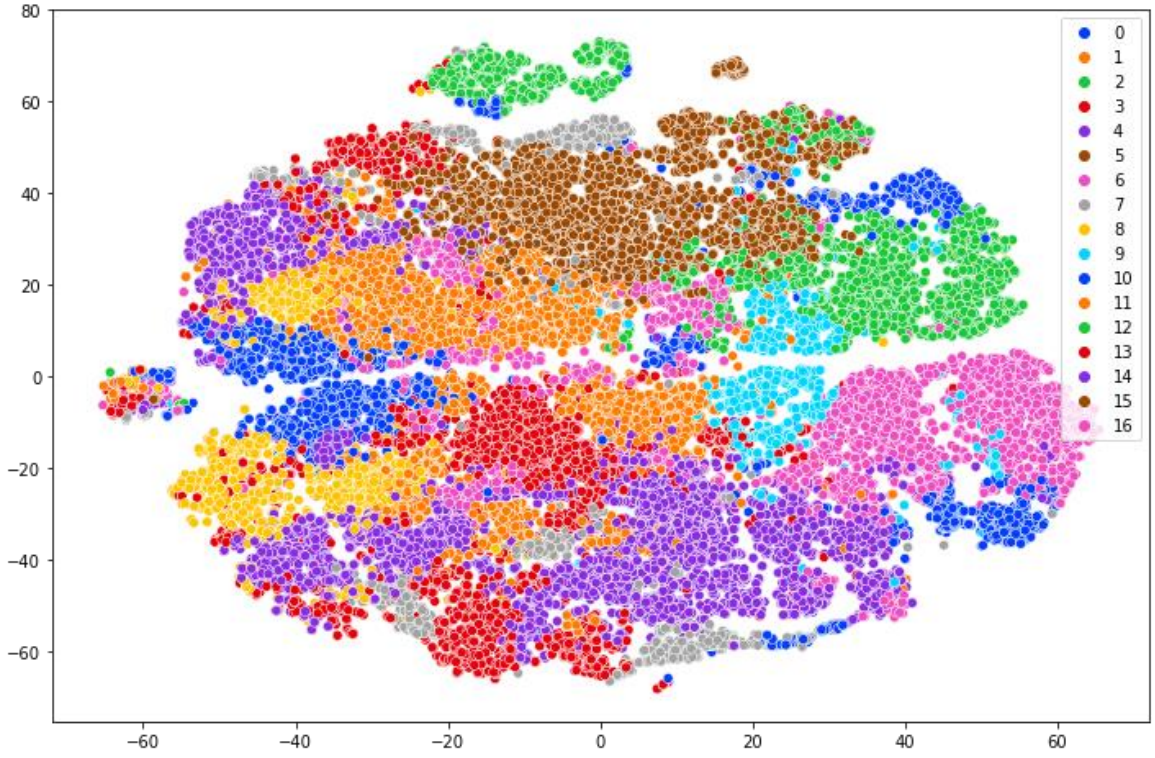
Min-max scaler



Normalizer



Standard_Scaler



با توجه به تمودار های به دست آمده بنظر انجام دادن Normalizer نتیجه بهتری خواهد داشت.

حال برای انجام recommendation نیاز است که داده های پلی لیست ورودی را Normalize کرده و سپس با توجه به مدل KMN سیو شده، predict را بر روی داده های جدید انجام دهیم. ابتدا پس از لود کردن مدل متوجه خواهیم شد که یوزر از کدام کلاستر ها بیشتر آهنگ گوش داده است، ۵تا کلاستر اول را در نظر گرفته و از هر کدام به صورت رندم از دیتاست اصلی آهنگ(که لیبل کلاستر ها در آن مشخص است) پیشنهاد میدهیم.

در پلی لیست ورودی ۱۰۰ آهنگ داده شده بود، پس از ران کردن مدل کلاستر بر روی این آهنگ ها، مشاهده میکنیم که از هر کلاستر به چه تعداد آهنگ توسط یوزر گوش داده شده است:

```
Counter({5: 40, 0: 30, 8: 19, 9: 9, 7: 1, 3: 1})
100
```

همانطور که میبینیم از کلاستر ۵ بیشترین تعداد آهنگ گوش داده شده است و به ترتیب کلاستر های ۰، ۸، ۹ و ...

برای TASK1، به ترتیب از کلاستر هایی که بیشترین تعداد آهنگ از آنها گوش داده شده است، ۵ آهنگ به صورت رندم برای هر کلاستر پیشنهاد میدهیم.

مثلا از کلاستر ۵ که ۴۰ آهنگ گوش داده شده بود، ۵ sample گرفته و به عنوان daily_mix1 خروجی میدهیم.

به همین ترتیب برای کلاستر های 7 , 9 ,8 , 0 نیز همینکار را تکرار کردیم تا daily_mix ها به ترتیب از ۱ تا ۵ به عنوان خروجی داده شده‌اند.

برای TASK2 با توجه به تعداد آهنگ های گوش داده شده در هر کلاستر، یک الگویی را پیش گرفتیم. از ابتدا از کلاستر شماره ۵، ۱۰۰ آهنگ به صورت sample گرفتیم و برای پیشنهاد آهنگ از کلاستر بعدی به اندازه ۲۰ تا کم کردیم. یعنی از کلاستر شماره 0، ۸۰ آهنگ، از کلاستر 8، ۶۰ آهنگ و .... به همین روال ادامه دادیم.

در انتها این آهنگ ها را به عنوان یک فایل CSV خروجی دادیم تا ترکیبی از همه کلاستر ها در پلی لیست خروجی نمایان شود.