به نام خدا

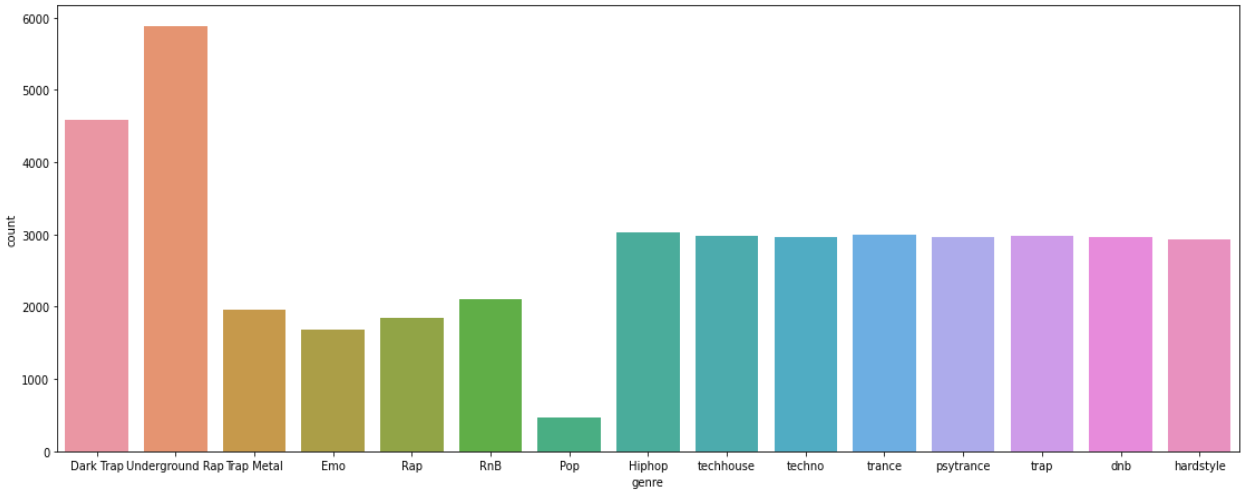
محمد زیاری – (97222047)

تاریخ: 19/ 09/ 1400

گزارش تمرین سری 3 **درس: مبانی علوم داده** **استاد درس: دکتر خردپیشه**

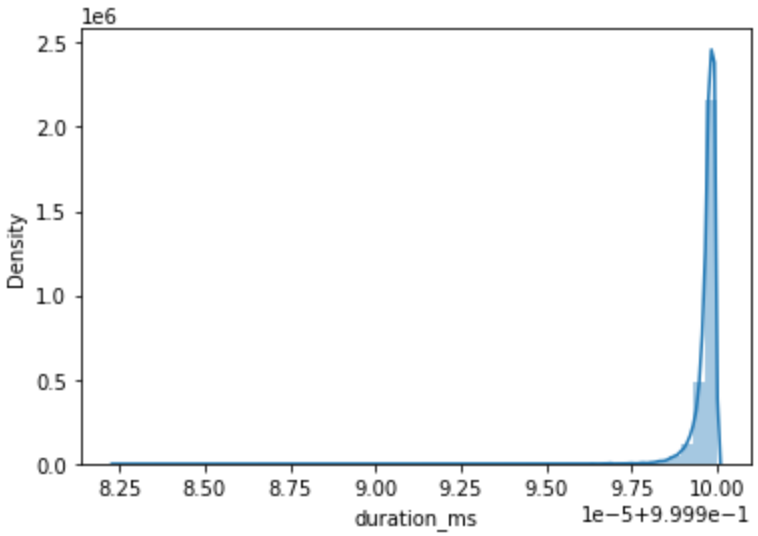
در این دیتاست ما با چند نمونه از آهنگ هایی که در اسپاتیفای ثبت شده اند مواجه هستیم و بر اساس آن ها ، اگر پلی لیستی به ما داده شد باید آهنگهایی از دیتاست خودمان به آنها پیشنهاد یا recommend بدهیم. برای این کار از روش clustering یا خوشه بندی استفاده میکنیم که در ادامه با آنها روبرو می شویم.

پس از خواندن داده ها از ککل ، آن را clean می کنیم. ابتدا نموداری از ژانرهای دیتاست موجود می بینیم که به شکل زیر است:

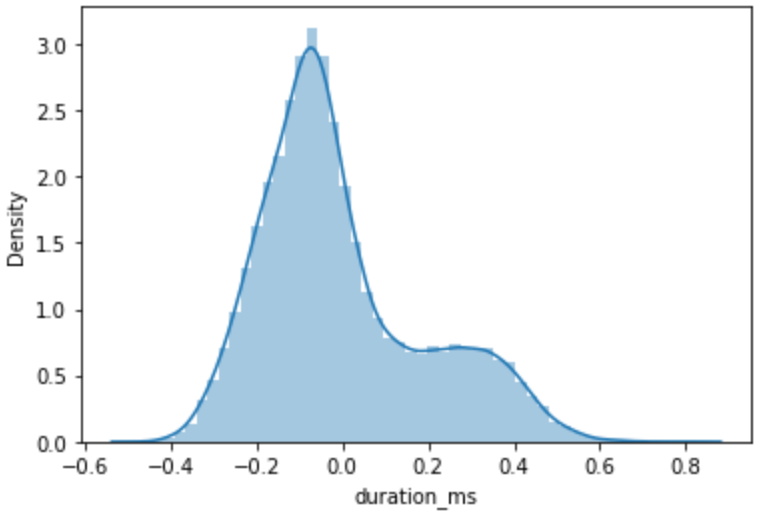


که مشخصا تنها 15 ژانر موجود است که نسبت به تعداد ژانرهای آهنگ های موجود در اسپاتیفای بسیار محدود و کم است. با این حال پس از دراپ کردن ستون **Unnamed: 0** برای اینکه کلاستر بندی را انجام دهیم ، ستون های 'type' ,'id', 'uri', 'track_href','analysis_url', 'song_name', 'title' را از دیتاست مان حذف می کنیم که ستونهایی مربوط به مشخصات آهنگ ها هستند و ستون هایی که برای سنجش مربوط به خوشه بندی به دردمان می خورد را در دیتاستی جداگانه نگه داشتیم.

در پردازش داده ها با چند مسئله روبرو هستیم که باید برای خوشه بندی بهترین کار را در هر روش انتخاب کنیم. اول از همه بررسی میکنیم تا داده ها را با استفاده از چه روشی نرمال کنیم. من اول از همه از متد normalized کتابخانه sklearn استفاده کردم. نمودار زیر نشان دهنده توزیع ستون duration_ms در دیتاست است.

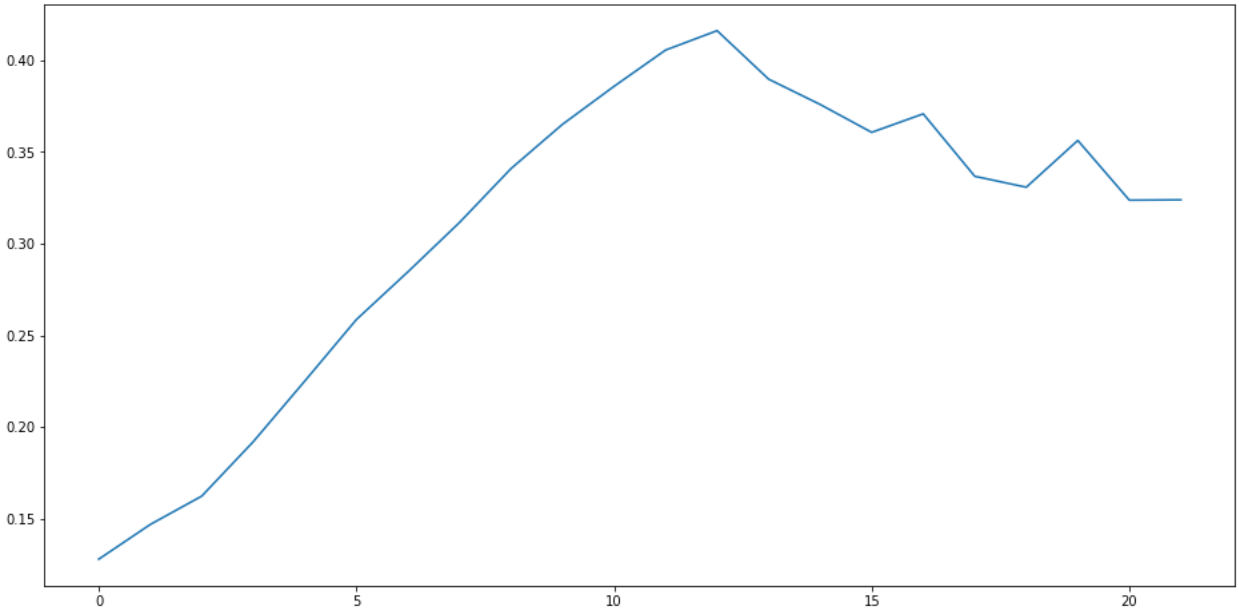


همانطور که مشخص است داده ها با اینکه توزیع نرمالی دارند ، بسیار به یکدیگر چسبیده اند که به علت داده های پرت است که این مورد بر روی پیشبینی ما نیز موثر است. پس ایده ای که به ذهن من رسید آن بود که داده ها را ابتدا با استفاده از standard scaler اسکیل کرده و پس از آن دوباره آنها را normalized کنیم.
نمودار پایین همان توزیع duration_ms این بار پس از انجام کارهای یاد شده می باشد که به نظر به ان شکل به هم نزدیک نیستند و میتوان پیشبینی بهتری ارائه داد.

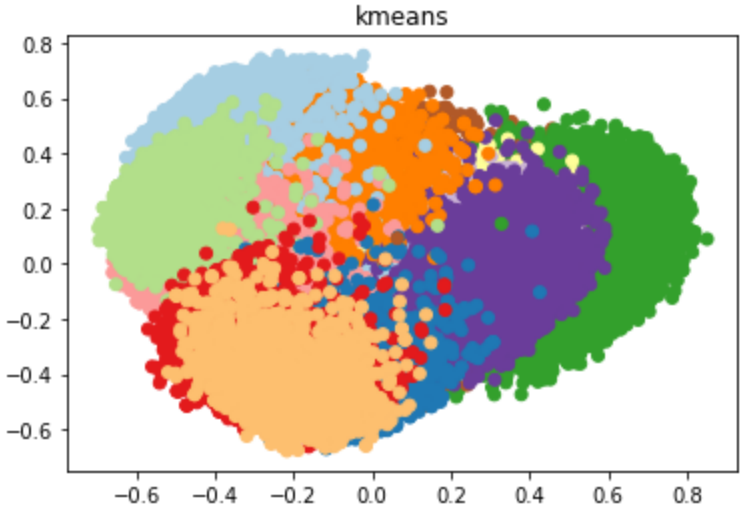


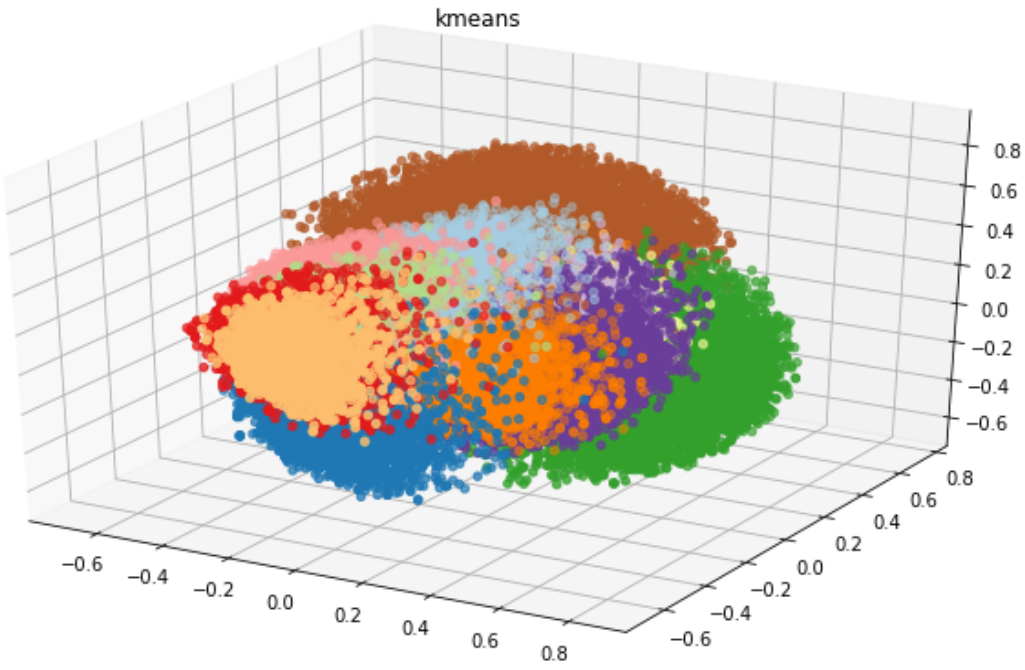
موضوع بعدی که در واقع باید قبل از نرمال کردن داده ها به آن بپردازیم عددی کردن داده های categorical است که در این دیتاست تنها داده کتگوریکالی که باید آن را مورد سنجش یا خوشه بندی قرار دهیم ، ژانر است .
اول از همه با استفاده از one hot coding هر یک از مقادیر ژانر را به ستونهای خاصی نسبت دادیم. پس از آن می خواهیم خوشه بندی را با استفاده از kmeans روی داده های مورد نظر انجام دهیم. برای انکه متوجه

شویم از چه تعداد خوشه استفاده کنیم از معیار سنجشی به نام silhouette_score بهره می بریم. نمودار پایین این مقدار بر حسب تعداد cluster هاست. (البته شماره خوشه ها باید به علاوه 3 شود زیرا در اندیس 0 ، 3 خوشه داریم در اندیس 1 ، 4 خوشه و الی آخر)

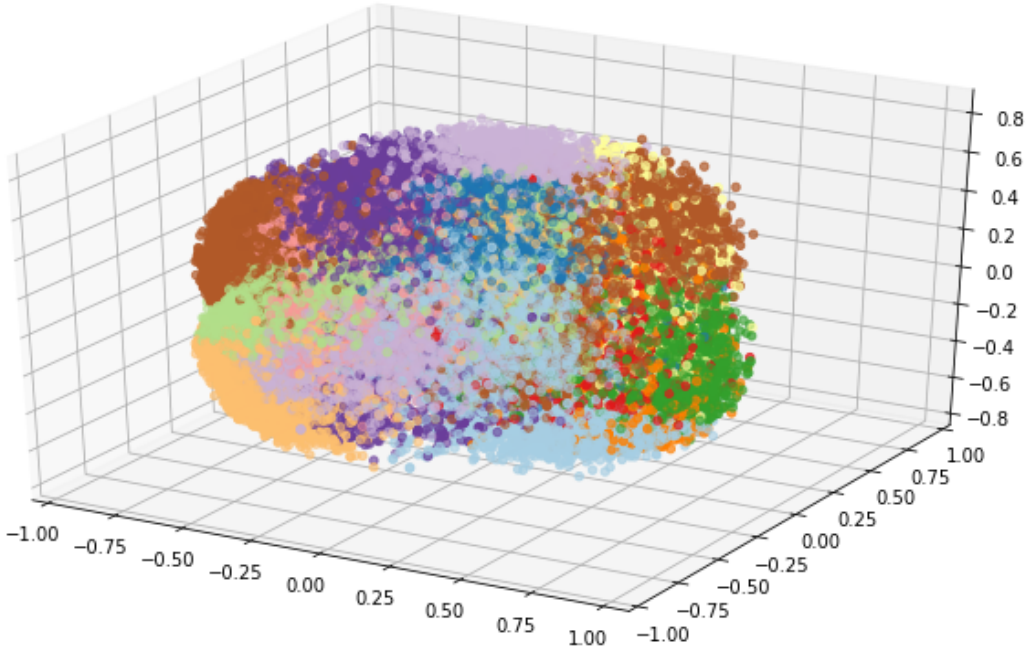


همانطور که مشخص است بیشترین میزان silhouette_score در i=12 است که مربوط به آن است که 15 خوشه داشته باشیم. نمودار خوشه ها با استفاده از تبدیل pca بر روی 2 بعد و سپس بر روی 3 بعد در شکل های زیر مشاهده می کنیم :



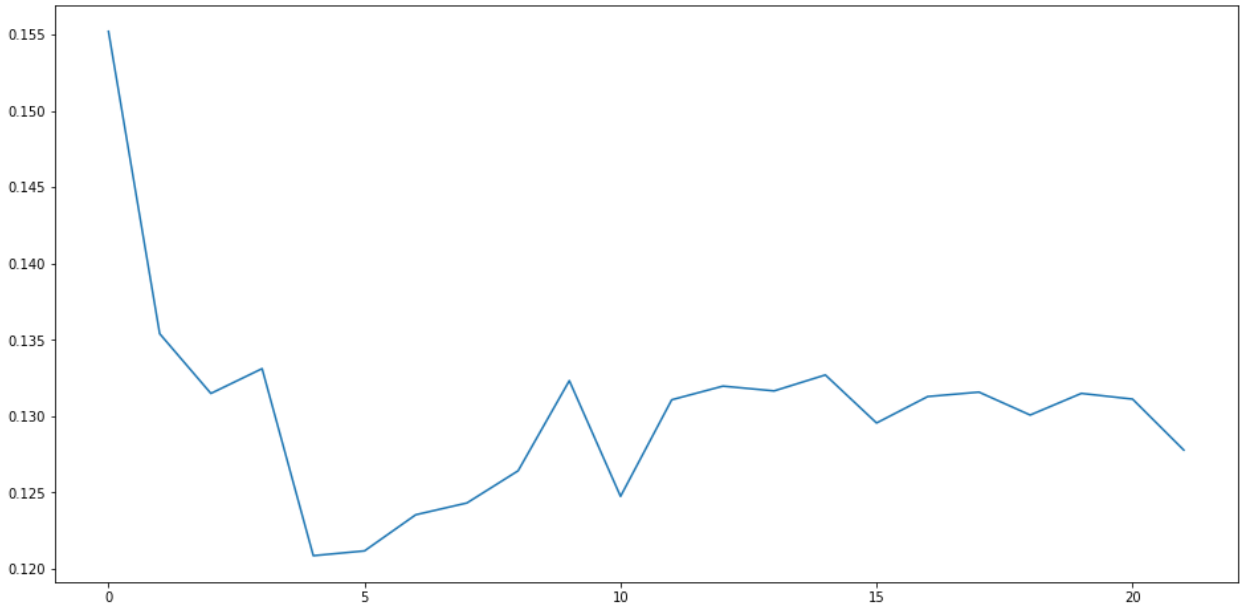


اما موردی که در این خوشه بندی قابل ذکر است ، آن است که با بررسی که روی خوشه ها انجام شده است ، clustering کاملا رابطه مستقیمی با ژانر دارد و در هر کلاستر معمولا یک ژانر قرار گرفته است که دلیل آن به نظر می تواند one hot کردن ستون ژانر است که تاثیر ستون های دیگر را بسیار کاهش داده است. راه حلی که برای این موضوع در نظر گرفتم ان است که label encoding انجام دهیم که البته این نوع عددی کردن نیز مشکلات مخصوص به خودش را داراست که از جمله آن این است که یک عدد بی مفهوم برای هر ژانر در نظر گرفته میشود و در واقع عدد یک سری از ژانر ها به یکدیگر نزدیک هستند در حالی که اصلا شبیه به هم نیستند. با این حال یک خوشه بندی با k means دیگر با این خصوصیت انجام دادیم که نکته قابل توجه آن است که این بار silhouette_score اصلا مقدار max در جایی که معنا و مفهوم خاصی داشته باشد وجود ندارد و بیشترین مقدار آن در 3 خوشه است که منطقی نیست که داده ها را 3 خوشه کنیم چون میخواهیم بر اساس خوشه ها recommend انجام دهیم. از این لحاظ من 25 کلاستر را در نظر گرفتم زیرا معیار سنجش پس از این مقدار به نظر با شیب بیشتری رو به کاهش داشت. شکل زیر نمایانگر pca در 3 بعد برای این خوشه بندی است :

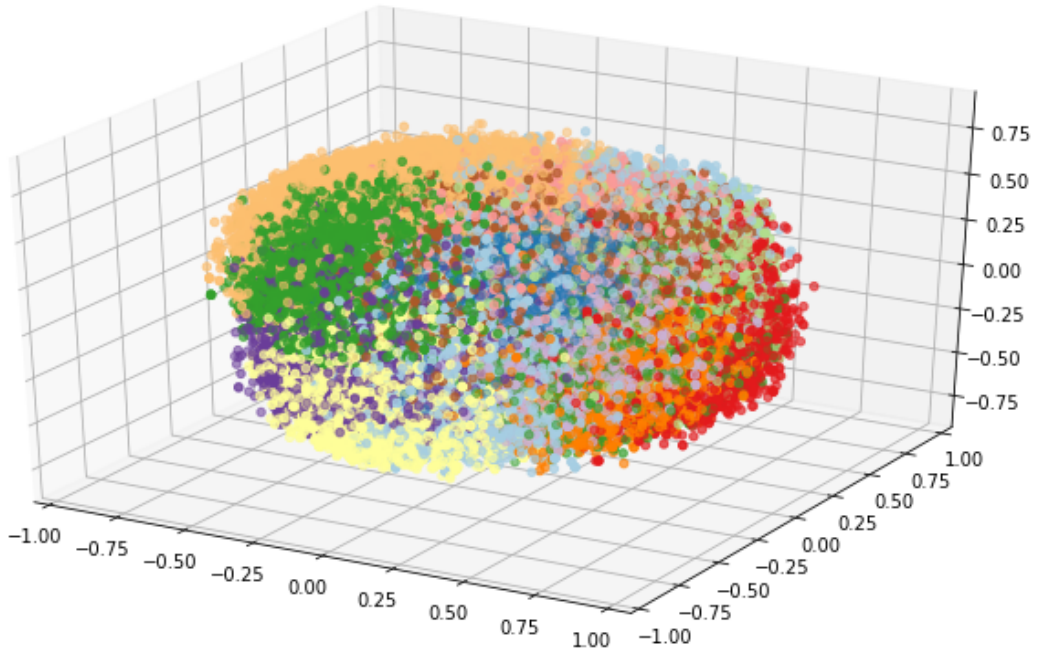


به نظر می آید خوشه بندی خیلی مناسبی نیست و تفکیک داده ها به خوبی در آن انجام نشده است. جدا از آن با توجه به اینکه از صدها ژانر موجود در اسپاتیفای تنها 15 ژانر آن در جدول موجود بودند و همینطور در api اسپاتیفای هم آهنگ ها به خودی خود ژانر ندارند و باید ژانر را از آرتیست دریافت کرد به نظرم منطقی تر به نظر میرسد که clustering را بدون ستون ژانر در نظر بگیریم تا بتوانیم پیش بینی بهتر و دقیق تری داشته باشیم.

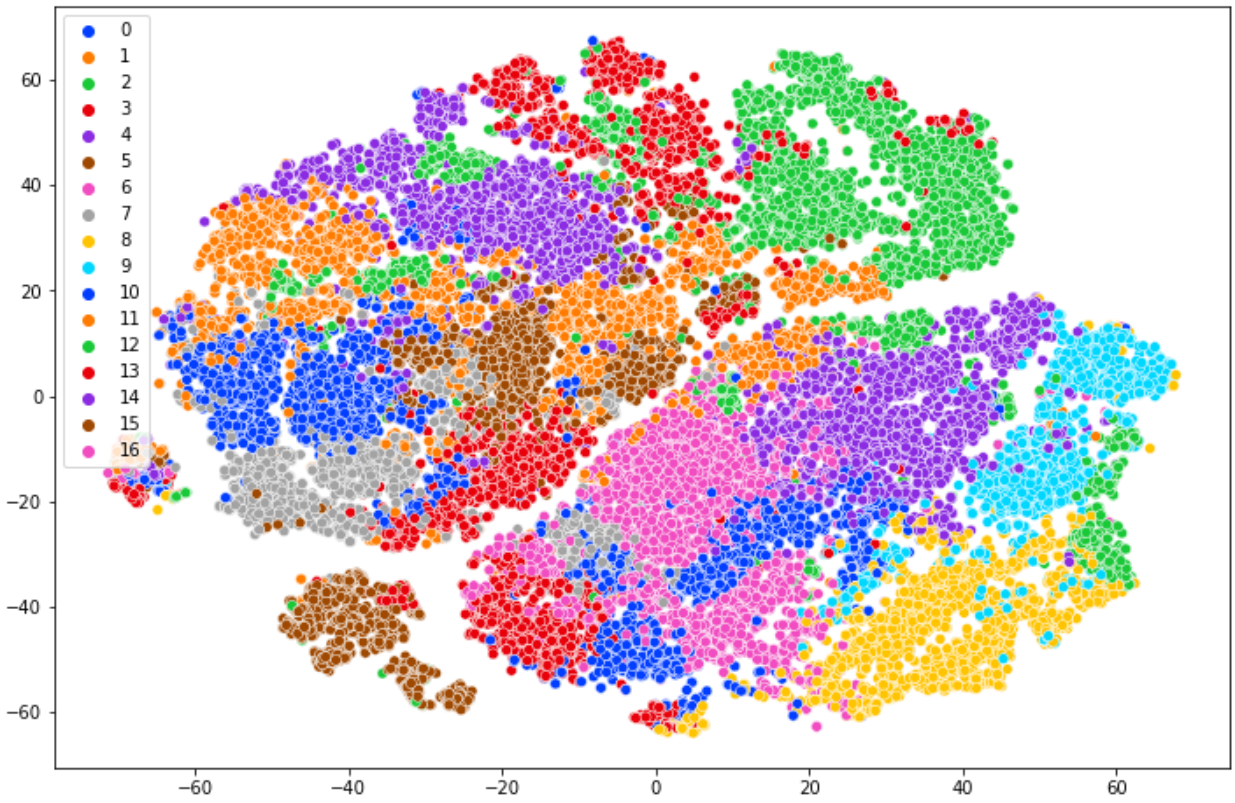
در زمانی که از ستون ژانر استفاده نکردیم و داده ها را با استفاده از ترنسفرمر یا تبدیل های استاندارد اسکیلر و normalized تبدیل کردیم ، میزان silhouette_score بسیار نوسانی عمل کرد. نمودار زیر نشان دهنده همین متغیر سنجش است که بار دیگر ذکر این نکته لازم است که هر ایندکس نشان دهنده کلاستر خود عدد به علاوه 3 می باشد زیرا کلاستر ها را از 3 آغاز کردیم.



میزان این معیار سنجش کاملا کاهشی است. اما برای مثال منطقی نیست که تنها 3 کلاستر داشته باشیم . برای همین i=14 که همان 17 خوشه را نشان می دهد که پس از آن میزان silhouette_score کاهش می یابد را در نظر گرفتیم. نمودار زیر pca آن را در 3 بعد نشان می دهد.

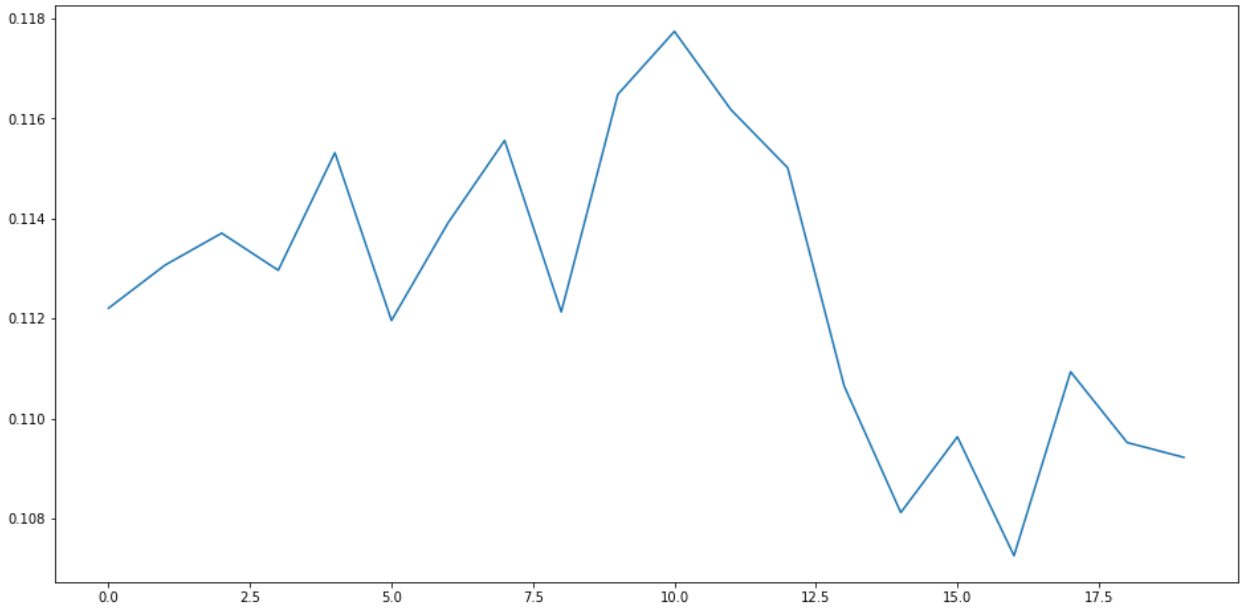


با استفاده از transformer ای به نام tsne نیز نمودار خوشه ها را در 2 بعد مشاهده می کنیم.

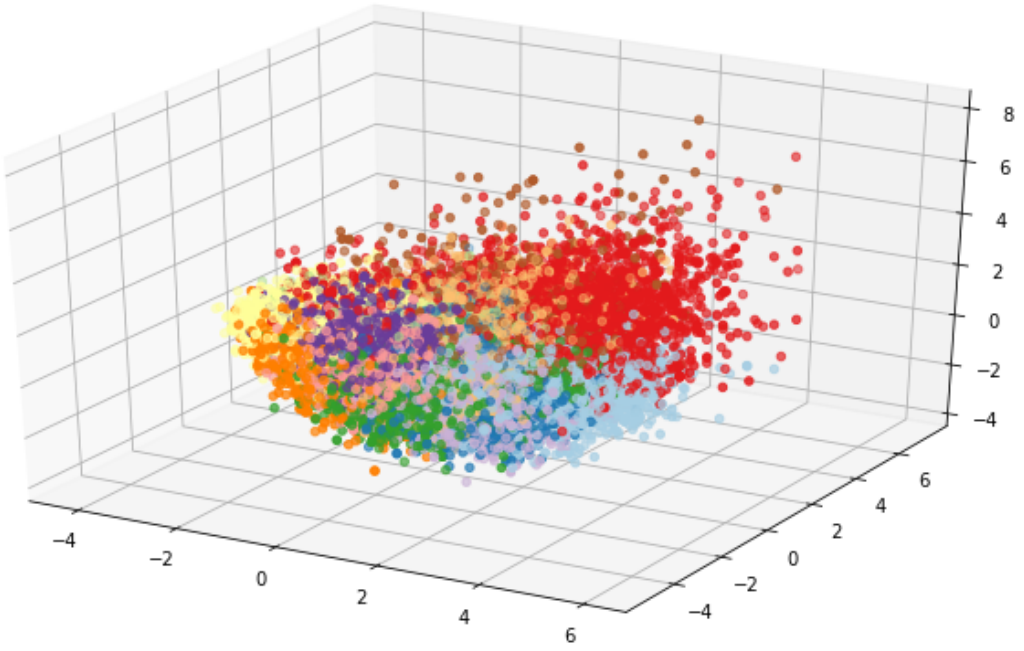


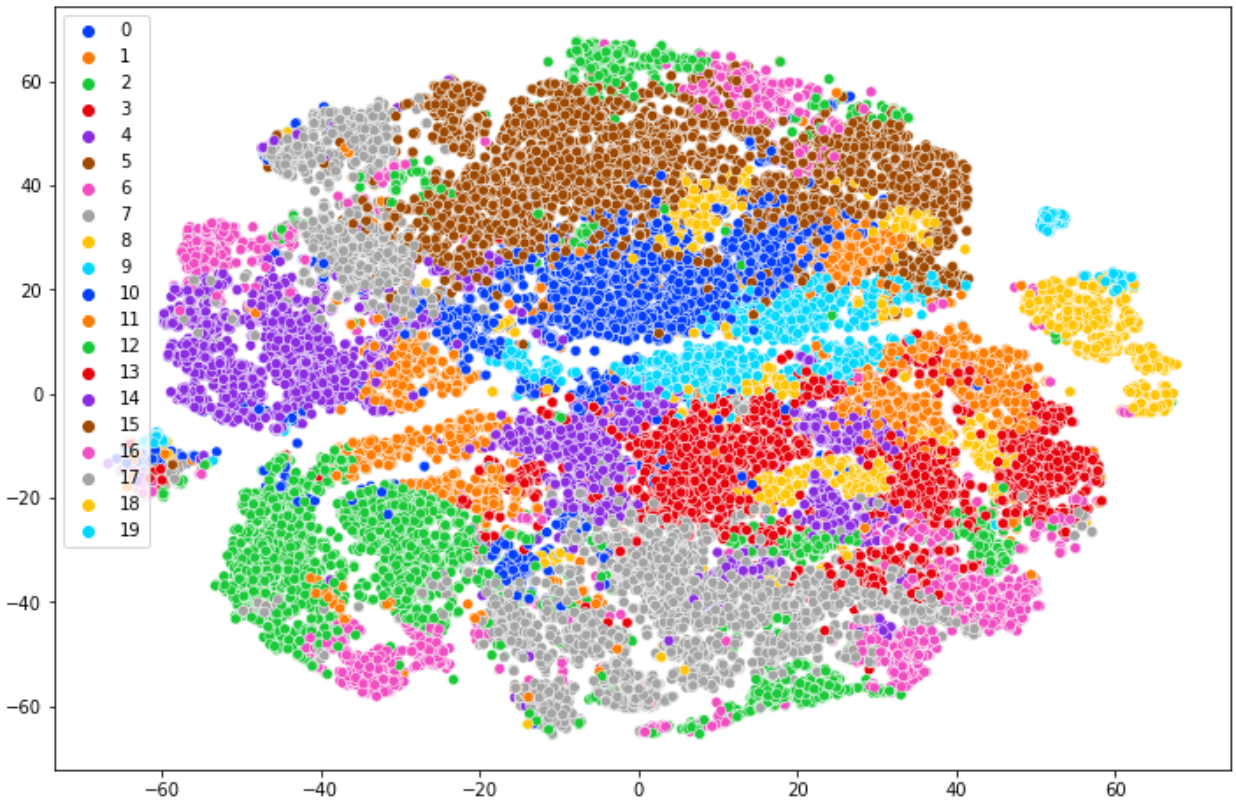
همانطور که مشخص است به نظر خوشه ها خیلی دقیق و درست از یکدیگر جدا نشده اند. کاری که به ذهن من برای انجام رسید آن بود که داده ها را با اسکیل و تابع های مختلف امتحان کنیم تا ببینیم در چه حالتی خوشه ها بهتر از یکدیگر جدا شده اند و پیشبینی دقیق تری خواهیم داشت.

اول از همه standard scaler را به تنهایی امتحان میکنیم. برای این داده نمودار silhouette_score را در 10 الی 25 خوشه میبینیم که قابل ذکر است این بار هر ایندکس تعداد خوشه 10 تا جلوتر از خودش را نشان می دهد.(یعنی i=0 -> cluster = 10)

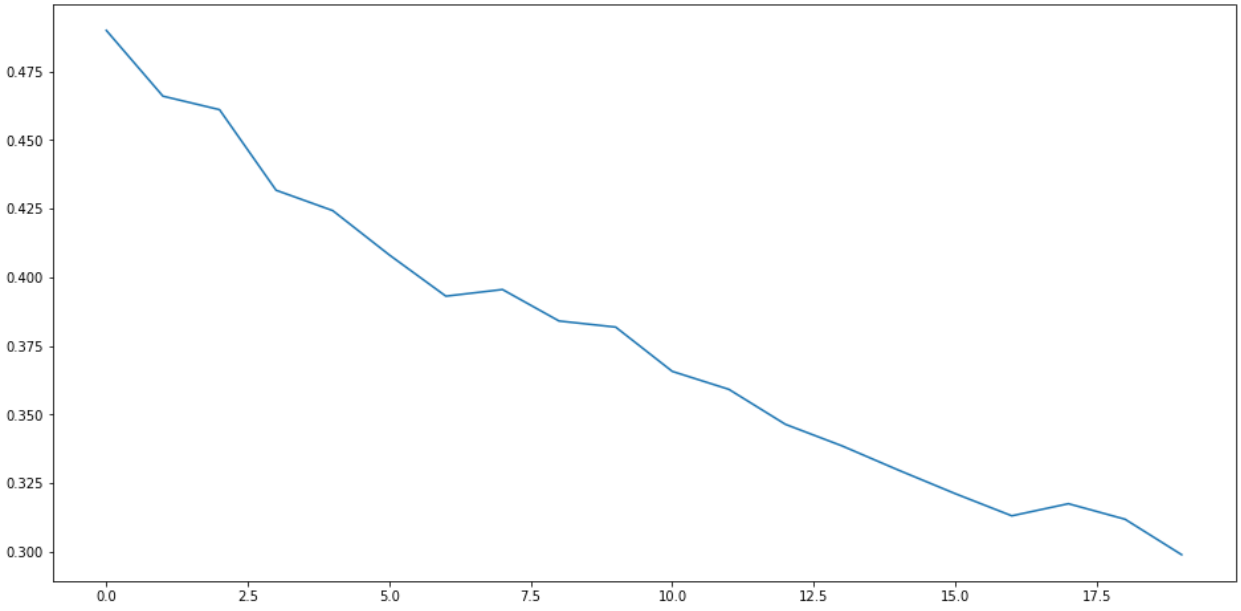


مشخصا بیشترین silhouette_score مربوط به i=10 یا همان 20 خوشه است. وقتی مدلمان را با 20 کلاستر فیت می کنیم نمودارهای زیر را در 3 و 2 بعد مشاهده می کنیم.

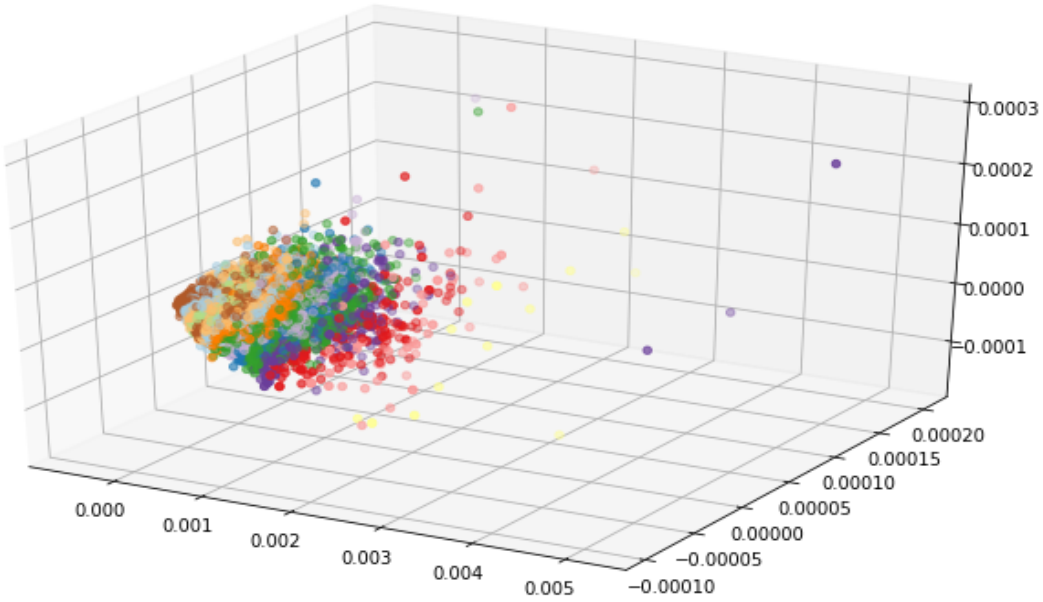


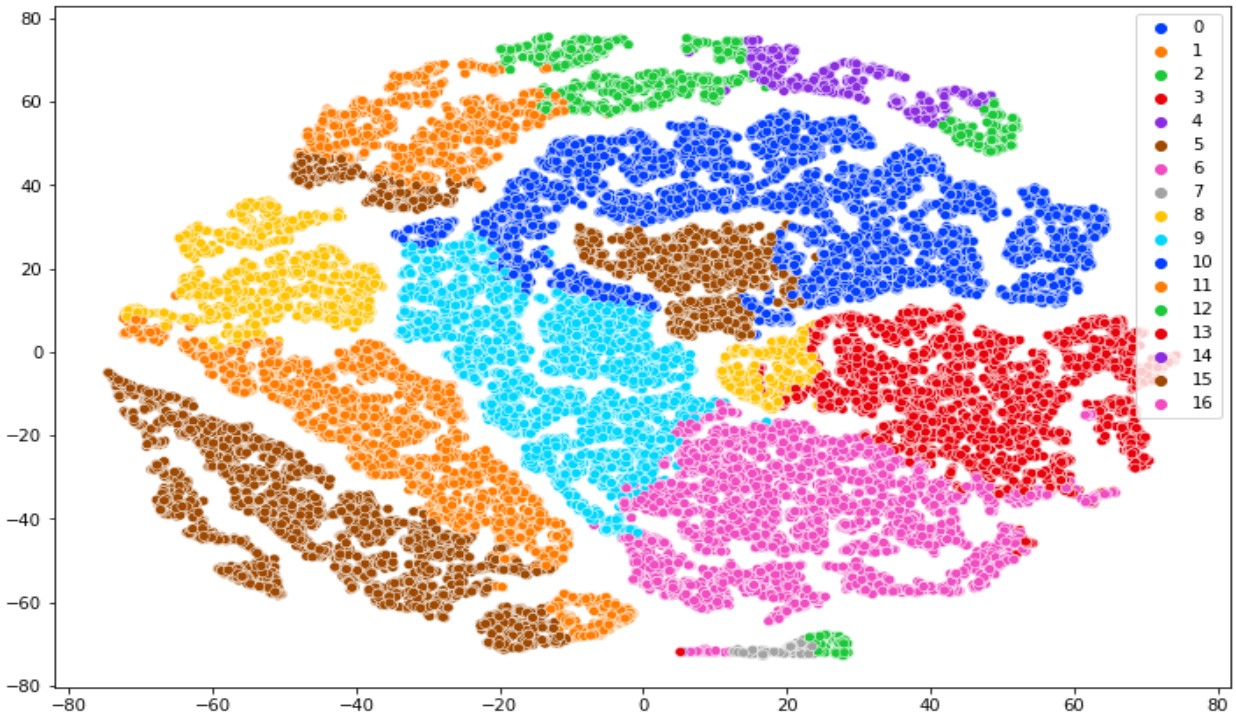


از روی نمودار ها و معیار سنجش این طور به نظر میرسد که همان حالت اولی که standard و سپس normalized شده اند پاسخ بهتری داشت. اما برای آنکه حالت های دیگر را نیز بررسی کنیم اینبار به سراغ تنها normalized کردن می رویم. جدول تغییر silhouette_score در آن به شکل زیر است.

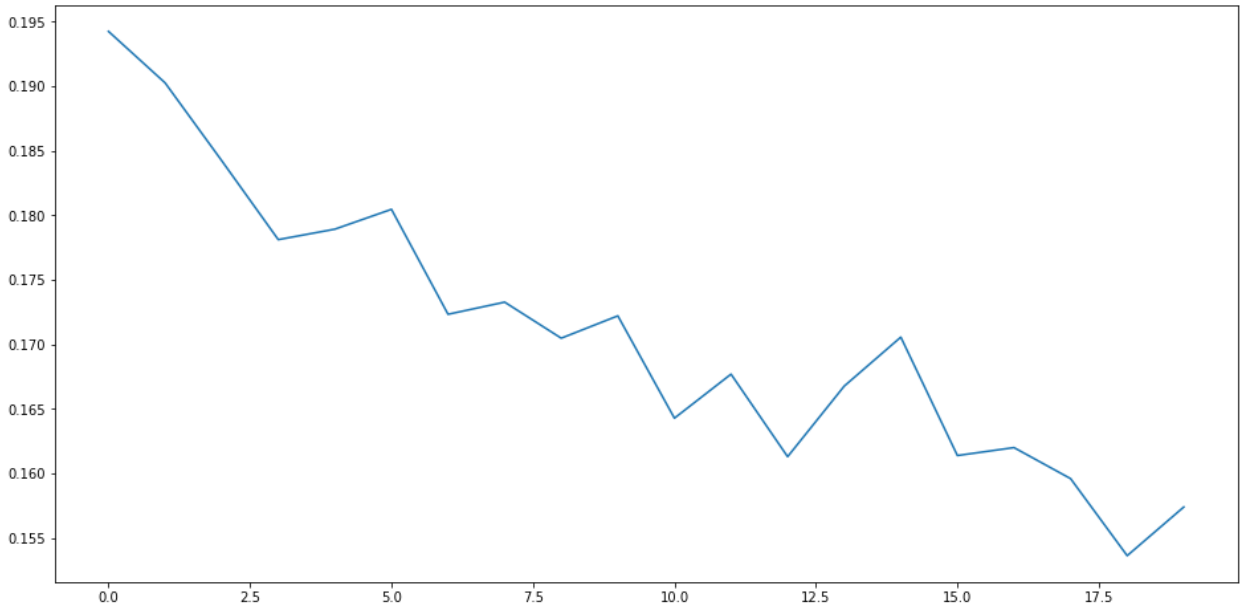


روند این معیار کاملا کاهشی است ولی مزیتی که نسبت به دیتاهای قبلی داشت این بود که مقدار سنجش ما به طور کلی بیشتر است. برای تشخیص بهتر مقدار کلاسترها را 17 در نظر گرفتیم چون از ان به بعد کمتر افزایش در مقدار معیارمان ایجاد شد. پس از آن نمودار های pca در سه بعد و tsne در دو بعد استفاده می کنیم و در زیر آنها را مشاهده می کنیم.

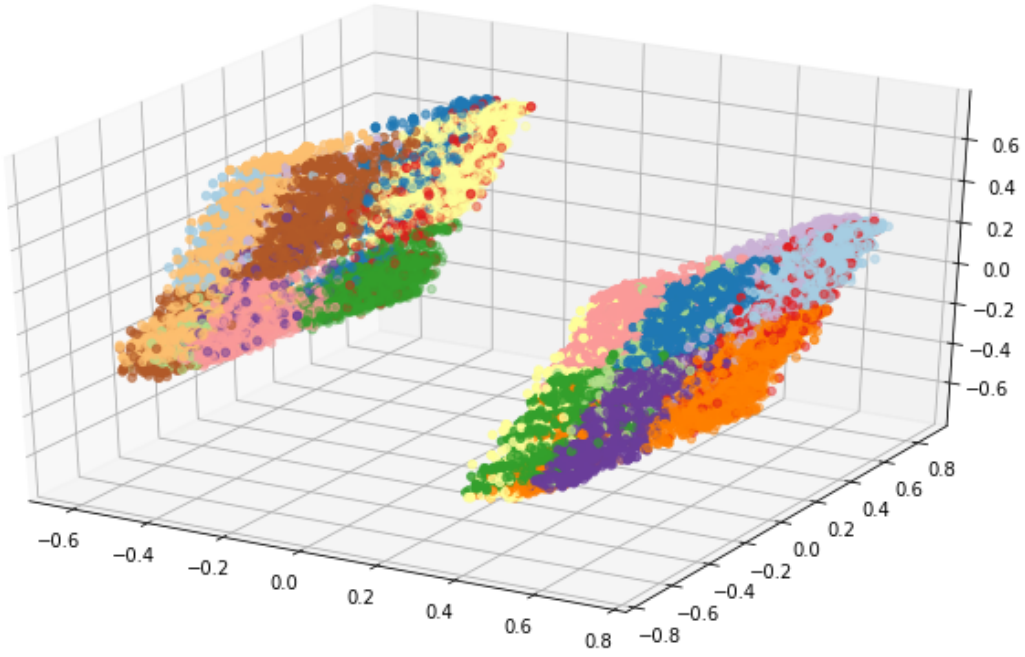


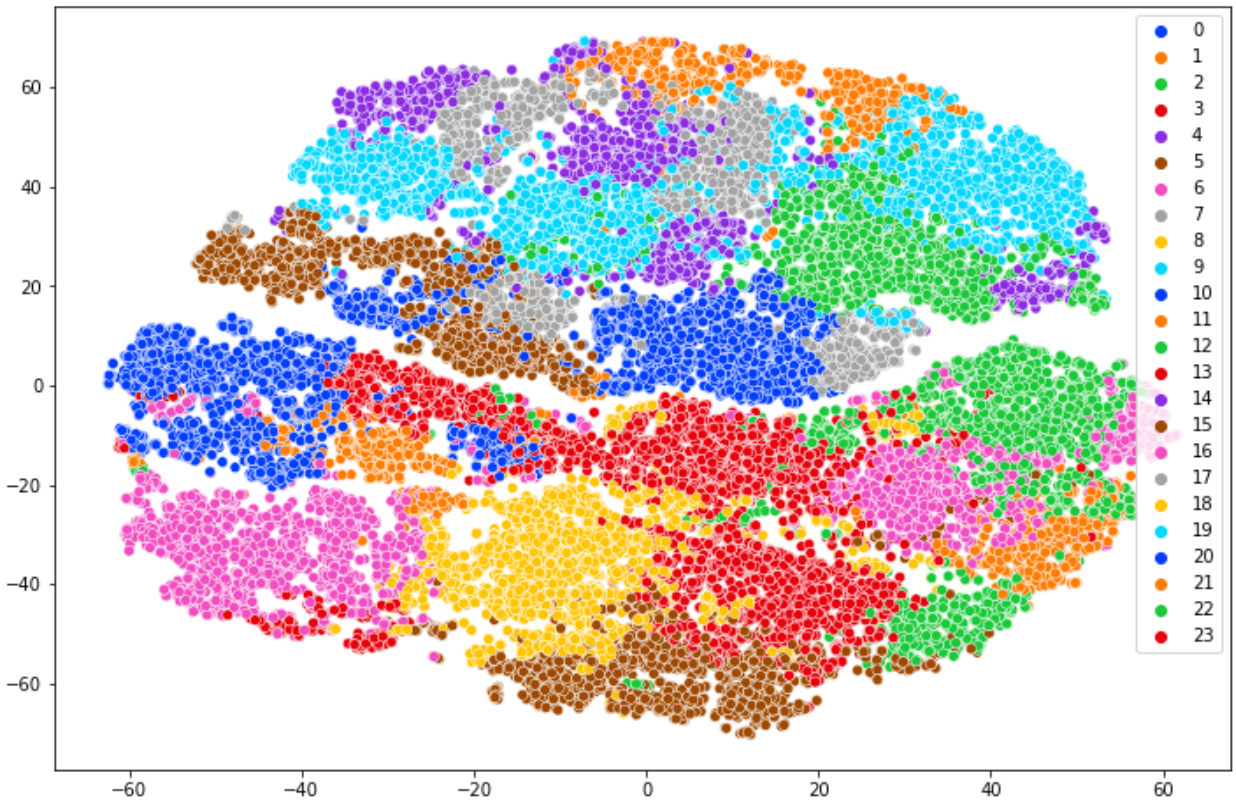


با اینکه به نظر نقص هایی در این کلاستر وجود دارد ولی به نظر بهترین مدل ما تا به حال همین مدل بوده است. اخرین نوع داده ای که در kmeans بررسی می کنیم اسکیل کردن با minmax scaler می باشد.
نمودار زیر نمایانگر silhouette_score در خوشه های 10 تا 30 عددی است.



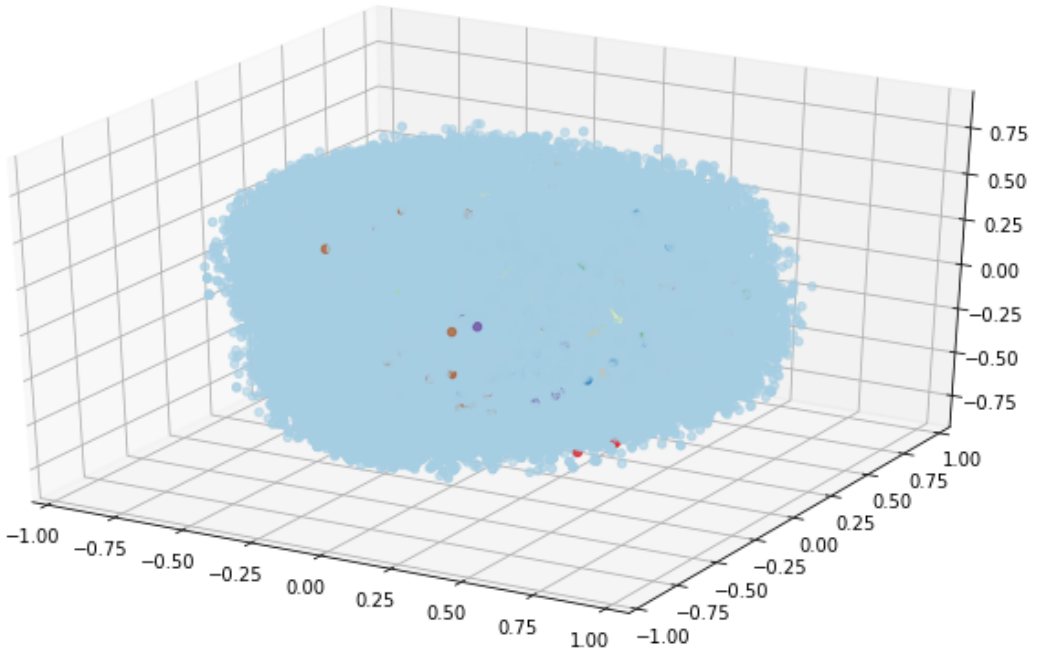
نمودار کاهشی است اما i=14 که نشان دهنده 24 کلاستر است از همسایه های خود مقدار بیشتری دارد پس این مقدار را انتخاب می کنیم. نمودار های pca در سه بعد و tsne در دو بعد را در ادامه مشاهده می کنیم.

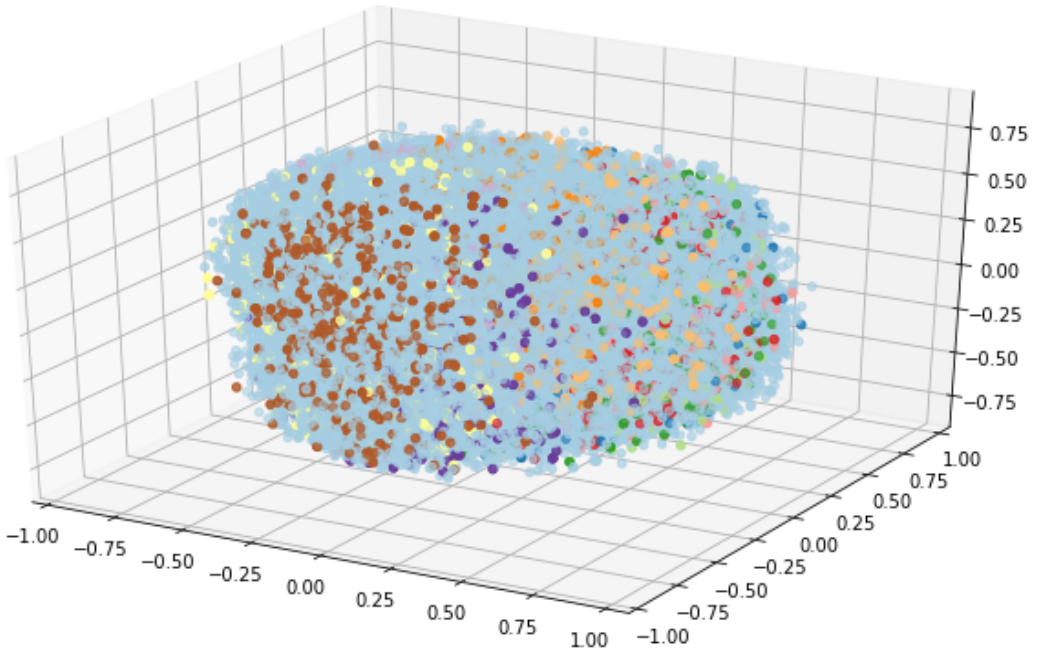




همانطور که در نمودار pca نیز مشخص است در این ترنسفورمر انگار 2 قسمت جدا از هم داریم که احتمالا به علت وجود یک یا چند feature است که تفاوت زیادی در آنها وجود دارد. به طور کلی به نظر این سبک کلاستر کردن نیز زیاد مناسب به نظر نمی رسد.

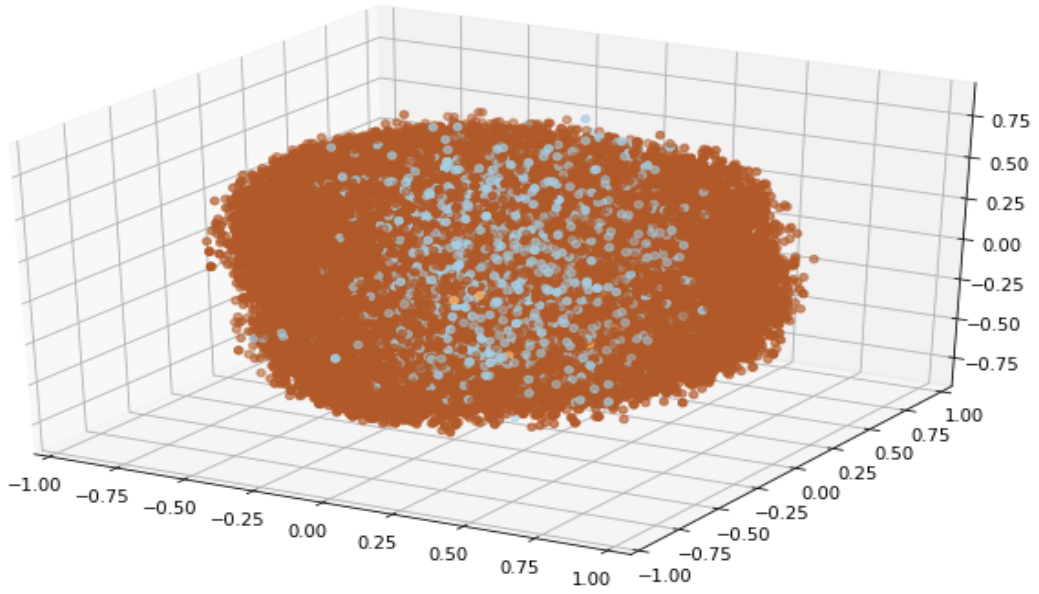
پس از این بررسی هایی که روی kmeans داشتیم حال نوبت آن است که خوشه بندی های دیگر را نیز مورد بررسی قرار دهیم. اول از همه DBSCAN را بررسی میکنیم که هایپر پارامترهایی را در خود جای داده است که از چندین سایت مختلف سعی بر این داشتم تا مقدار بهینه برای خوشه بندی را پیدا کنم ولی مثمر ثمر نبود. برای مثال در این لینک توضیح داده شده و من از eps های مختلف بررسی را انجام دادم اما هربار مقدار silhouette_score منفی یا بسیار بسیار پایین بود. پس به صورت دستی مقادیر هایپرپارامترها را تغییر دادم تا به نمودار زیر با pca در 3 بعد رسیدیم. در ابتدا نموداری با هایپرپارمترهای , eps = 0.4 min-samples = 7 و سپس eps = 0.1 , min-samples = 2 را میبینیم.



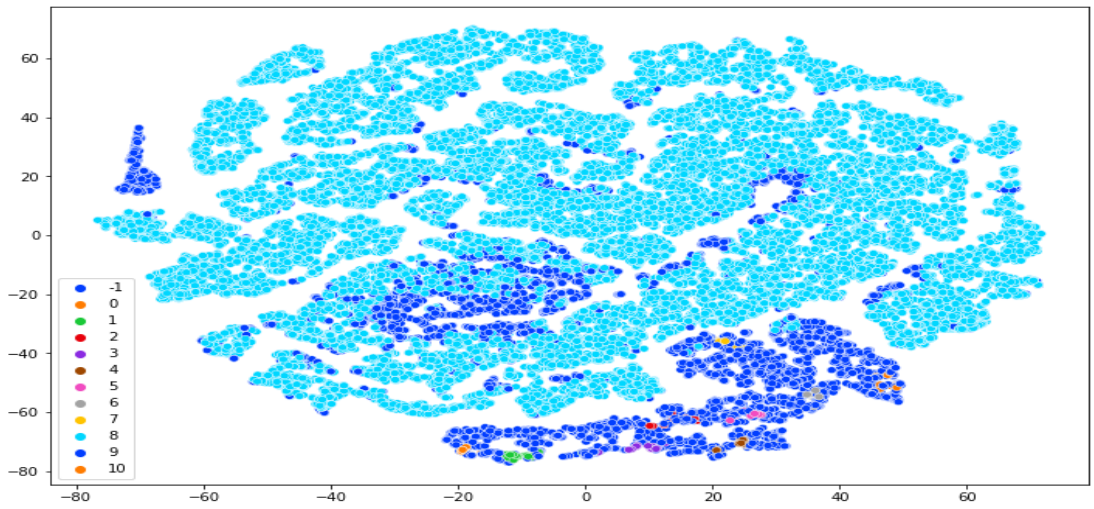


هر چند دومی اندکی واضح تر و بهتر به نظر میرسد اما هیچکدام از این خوشه بندی ها به هیچ عنوان قابل قبول و قابل استفاده نیست و من تریج دادم اصلا از این روش استفاده نکنم.

روش بعدی که از ان استفاده شده است hdbscan است. در این روش پارامترهای ما نوع فاصله است که چطور محاسبه شود و دیگری کمترین مقداری که در هر کلاستر باید جای بگیرد. اولین بار داده ای که هم استاندارد اسکیلر روی ان صورت گرفته و هم نرمال شده است را مدل کرده ایم که تنها 3 کلاستر تبدیل می شد و نتیجه خوبی نداشت . نمودار ان را در شکل زیر میبینیم.

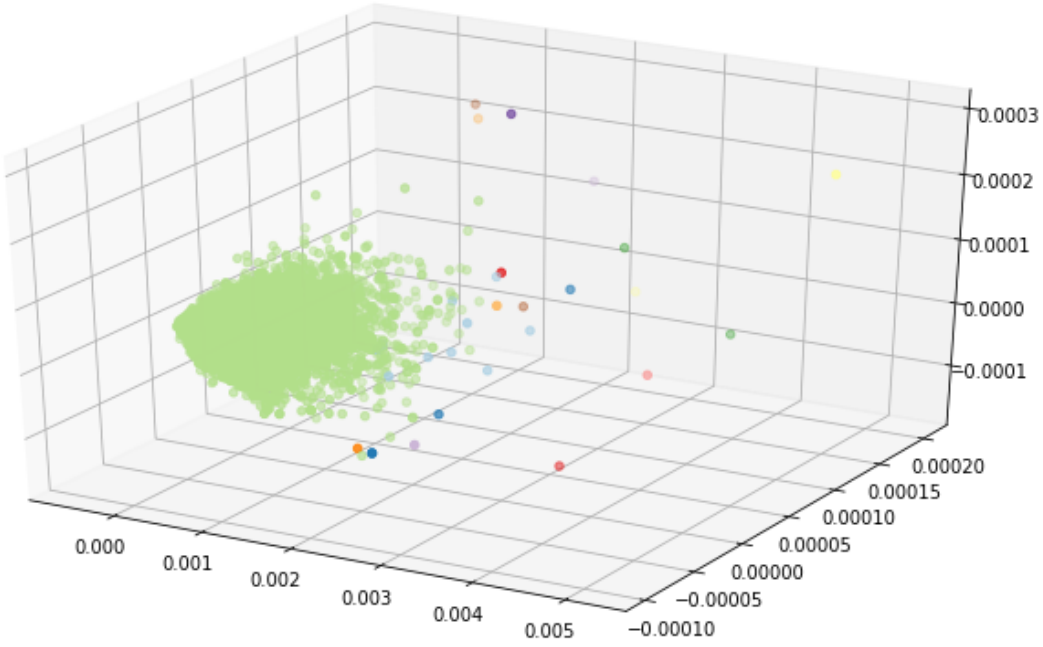


اما در قدم بعدی داده هایی که تنها normalized شده اند را بررسی می کنیم. در این داده ها اگر زمانی که min_cluster_size مقدار 10 را داشت تعداد کلاستر ها در حدود 40 عدد بود . در نتیجه مقدار ان را به 20 افزایش دادم و همینطور مقدار فاصله با رابطه اقلیدسی به دست می امد که نمودار زیر خوشه بندی انها در 2 بعد را نمایش می دهد.



این نمودار به هیچ وجه قابل قبول نیست. حتی زمانی که فاصله را به صورت منهتن تغییر دادم هم نمودار جالبی به دست نیامد و کلاستر بندی به این شکل بود که چند کلاستر بسیار داده های کمی را دارند و 2 الی 3 کلاستر باقی داده ها را در بر گرفته اند. پس این خوشه بندی نیز مورد استفاده قرار نمی گیرد.

در اخر هم از خوشه بندی Hierarchical استفاده کردم که این خوشه بندی نیز اصلا قابل استفاده نیست. در این خوشه بندی linkage قرار داده شده است که زمانی که خواستم مقدار ward را برای ان قرار دهم به سرعت کرش میکرد و قابل استفاده نبود برای هر تعداد کلاستری که به نظرم دلیلش داده زیاد ماست. زمانی که این پارامتر را single تعریف کردم با هر تغییری در بقیه پارامترها هیچ نتیجه خوبی حاصل نشد و به نظر می اید اکثر داده ها در یک کلاستر جای میگیرند. نمودار زیر نمودار pca در 3 بعد را نشان می دهد.



از اینجا به بعد شروع به جمع بندی و recommend کردن اهنگ ها می کنیم. مدلی که از ان برای پیشنهاد دادن استفاده میکنیم kmeans روی داده هایی است که تنها normalized شده اند. تعداد خوشه های ما نیز 17 تاست که به صورت رندم از آنها انتخاب میکنیم. دلیل انتخاب کلاستر بیشتر نیز همین علت بود.

پس از گرفتن فایل پلی لیست مورد نظر و نرمال کردن آنها به همان شکلی که داده اصلی نرمال شده بود ، با توجه به مدلی که ذخیره شده است کلاستر یا خوشه هر آهنگ در پلی لیست را با توجه به متغیراتش پیشبینی میکنیم.

5 پلی لیست با نام های mix1 , mix2 , mix3 , mix4 , mix5 که هرکدام پلی لیستی از یک کلاستر منحصر به فرد بوده اند که به صورت رندم از کلاستر هایی که بیشترین نقش در پلی لیست ما را داشتند انتخاب شده اند و در فایل تحویلی وجود دارد.

خروجی دیگری که مدنظر بود پلی لیستی از همه کلاسترها بود که من اینطور فرض کردم که هر چند تا آهنگ در یک کلاستر قرار بگیرد ، به همان اندازه اهنگ از آن کلاستر به ما پیشنهاد شود. این فایل نیز با نام mix all در خروجی ما مشخص شده است.