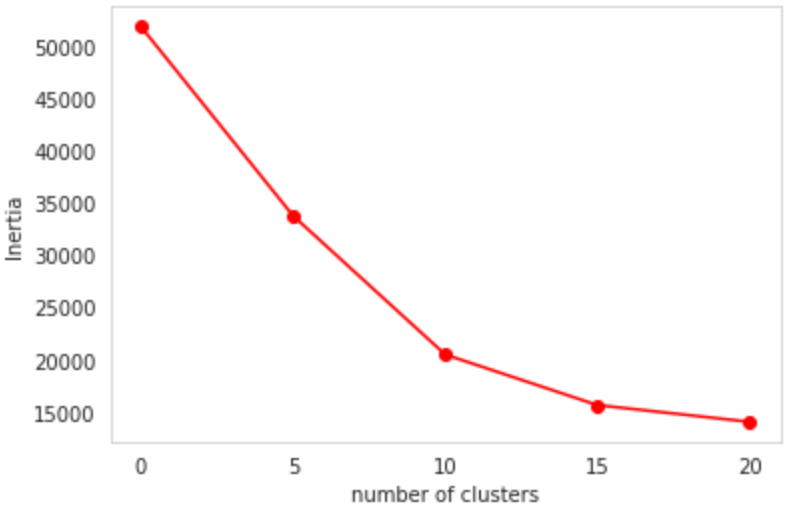
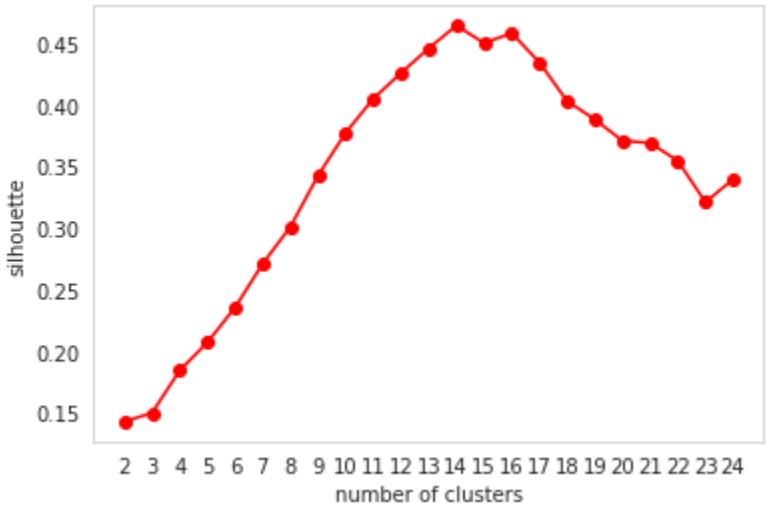
گزارش پروژه سوم علوم داده

محمدرضا صیدگر-97222055

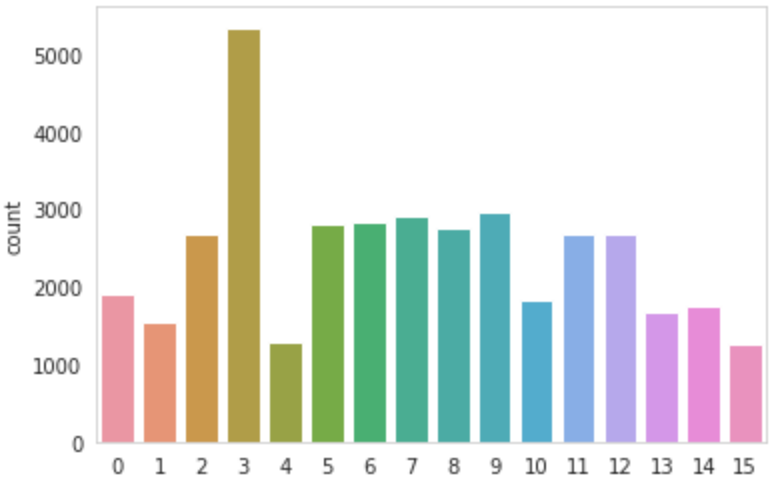
این پروژه اجرای یک recommender system است بر روی داده های موسیقی که دیتا ست ما مجموعه ای از ویژگی های آهنگ های مختلف موجود در اسپاتیفای است.
اول از همه داده ها را که میخوانیم باید بررسی کنیم که چه ویژگی هایی برای ما مهم است و میخواهیم روی آنها مدل ایجاد کنیم و بعد از آن ویژگی های غیر ضروری را پاک کنیم.
'song_name' , 'Unnamed: 0' ,
'title','type','id','uri','track_href','analysis_url','time_signature','mode'
بعد از بررسی این ستون ها را از داده ها پاک کردیم چون اکثرشون فایده ای برای تحلیل موسیقی نداشتند و 'time_signature','mode' این 2 ستون هم به این خاطر حذف شدند چون نرمال نبودند و داده ها را چند دسته میکردند.
سپس داده های outlier را حذف کردیم و کل دیتاست را scale کردیم به روش minmax که به این صورت است که هر داده x تبدیل میشود به x-min/min-max که min و max مربوط به کوچکترین و بزرگترین داده در آن ستون است.
در بخش اول تحلیل ستون genre را حذف نکردیم چون بر این فکر بودم که میتواند این ستون به ما کمک کند در رابطه پیشنهاد موسیقی چرا که خود من به شخصه یک و دو نوع موسیقی با ژانر خاص را می پسندم اما حال ببینیم که این کار چه نتیجه ای را به همراه داشت.
چون داده های genre داده های categorical بودند و مدل ها این داده ها را متوجه نمی شوند پس میبایست که عددی شوند پس 2 نوع سیاست را در پیش گرفتیم. سیاست اول این بود که این ستون را one hot encoding کنیم و سیاست بعدی این بود که label encoding کنیم. در هر کدام از این 2 سیاست 3 مدل کلاستر بندی dbscan , hierarchical , kmeans اجرا کردیم که در ادامه نتایج هرکدام را خواهیم دید:
حالت اول (one hot encoding , kmeans):
روی داده ها کلاستر بندی kmeans را اجرا کردیم اما از مشکلات kmeans این است که تعداد کلاستر را خودمان باید تعیین کنیم پس برای کلاستربندی های مختلف از 2 تا 25 کلاستر شاخص های inertia و silhouette را بررسی کردیم. Inertia میزان خوشه‌بندی یک مجموعه داده توسط K-Means را اندازه‌گیری می‌کند. با اندازه‌گیری فاصله بین هر نقطه داده و مرکز آن، مجذور کردن این فاصله و جمع کردن این مربع‌ها در یک خوشه محاسبه می‌شود.اما امتیاز silhouette معیاری است که برای محاسبه خوب بودن یک تکنیک خوشه‌بندی استفاده می‌شود. مقدار آن از 1- تا 1 متغیر است. که اگر منفی باشد یعنی کلاستر بندی خوبی نداشتیم و اگر هرچه به 1 نزدیک شود یعنی کلاسترینگ بهتر بوده.
اینرسی برای تعداد کلاستر مختلف به شکل زیر شد که هرچی تعداد کلاستر بیشتر شود بهتر شد.



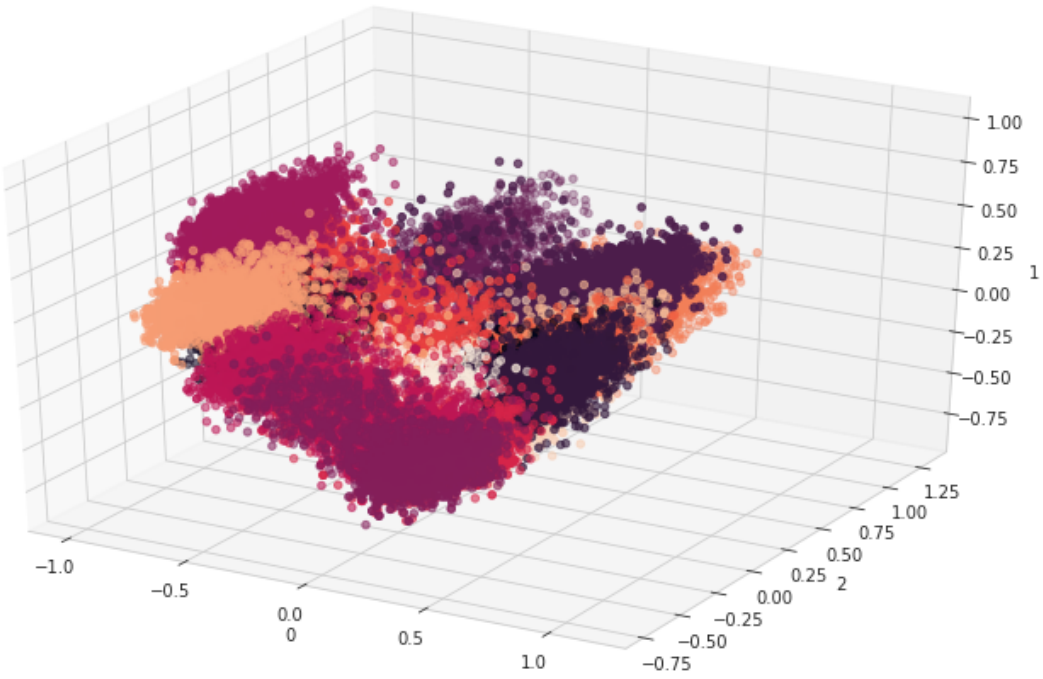
معیار silhouette هم برای تعداد کلاس های مختلف به شکل زیر شد که حدودا 14 تا 16 کلاستر بهترین حالت است.



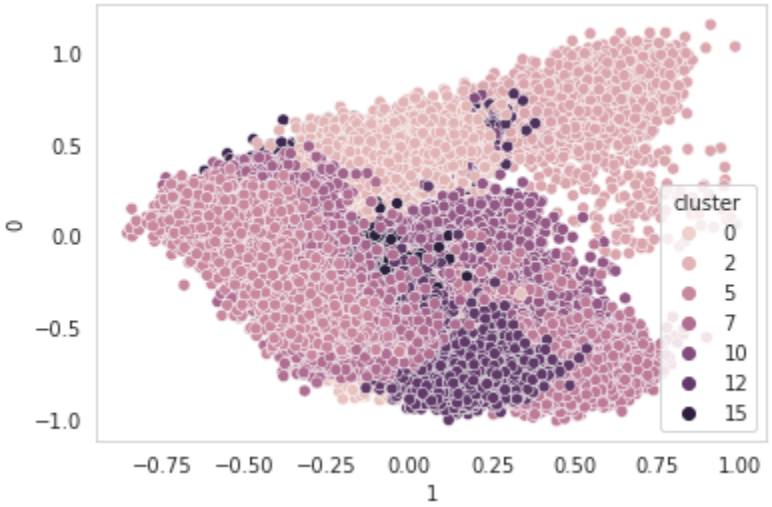
که در نهایت 16 کلاستر رو به عنوان مدل نهایی امتحان کردم که تعداد داده های هر کلاستر به شکل زیر شد



بعد از کلاستر بندی بررسی کردم در هر کلاستر داده هایی که وجود دارند از چه ژانر های مختلفی اند که فهمیدم در هر کلاستر 1 و نهایتا 2 ژانر وجود دارند که نشان میدهد one hot کردن ژانر به شدت تاثیر این را زیاد کرد چون ماتریس بسیار ستون هایش زیاد می شود و داده های هر ژانر از هم فاصله زیادی میگیرند پس منطقی این میشود که هر ژانر یک کلاستر شود.

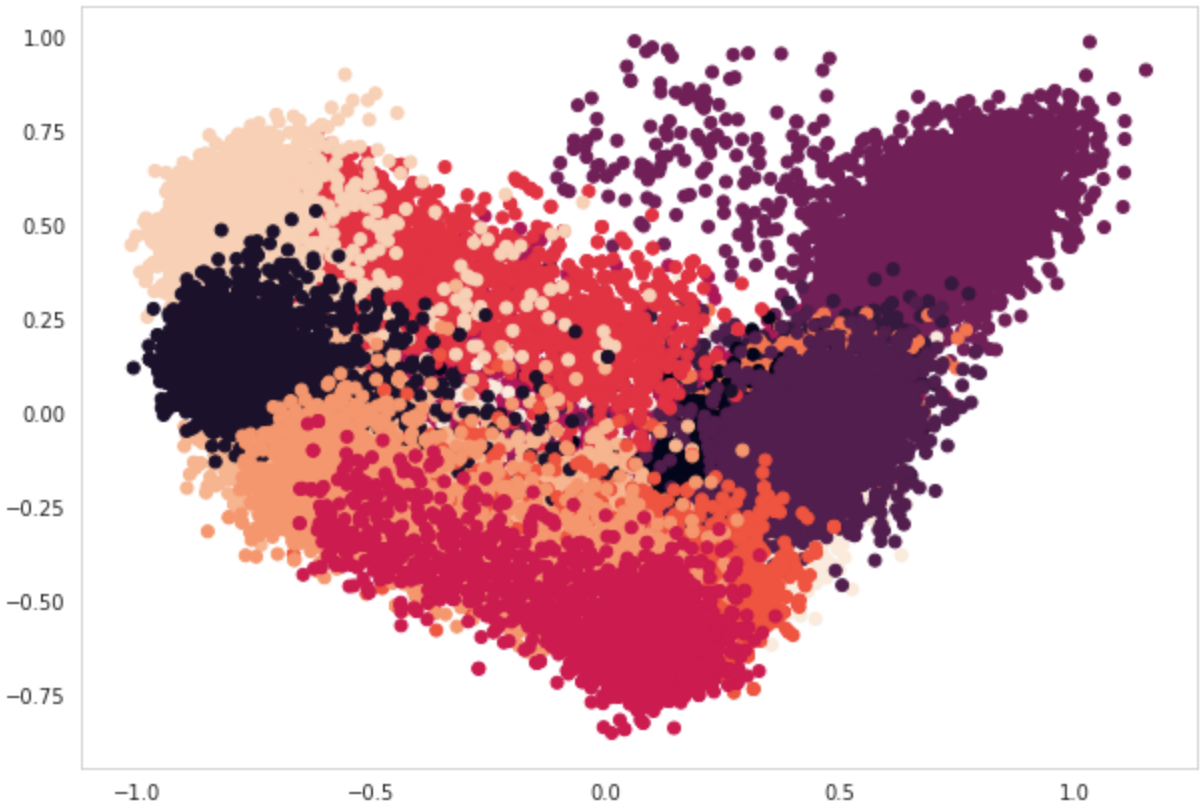


این نمایش داده ها و کلاستر بندی آنها در 3 بعد است.

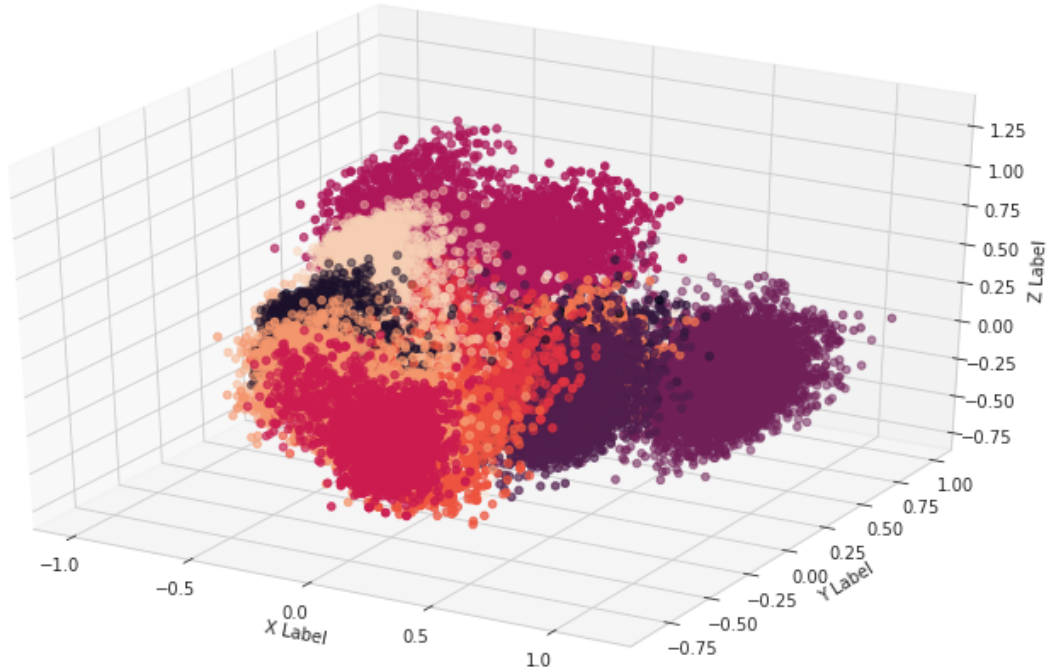


و این هم نمایش داده ها در 2 بعد است.

حالت دوم (one hot encoding , hierarchical clustering):
برای کلاستر بندی hierarchical هم تعداد کلاستر 15 رو گذاشتم با linkage single که فاصله بین کلاستر ها کمترین فاصله را می گیرد.

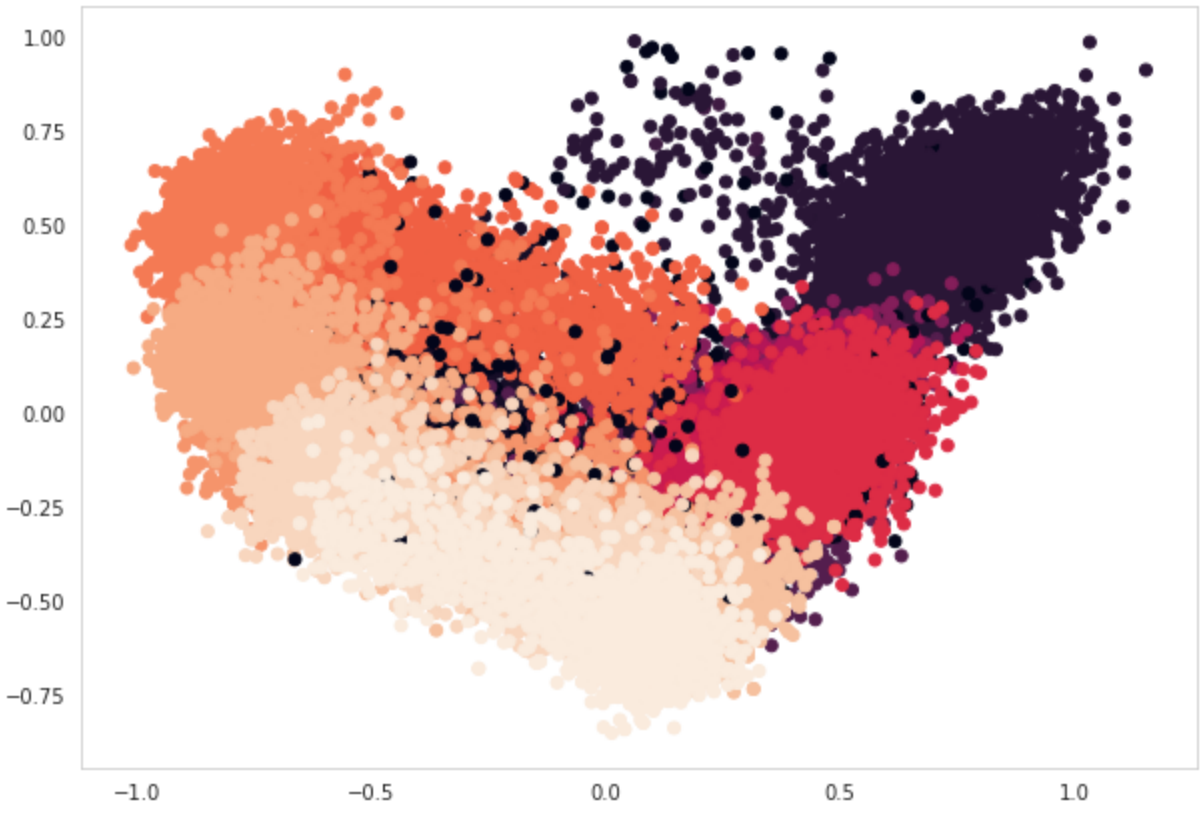


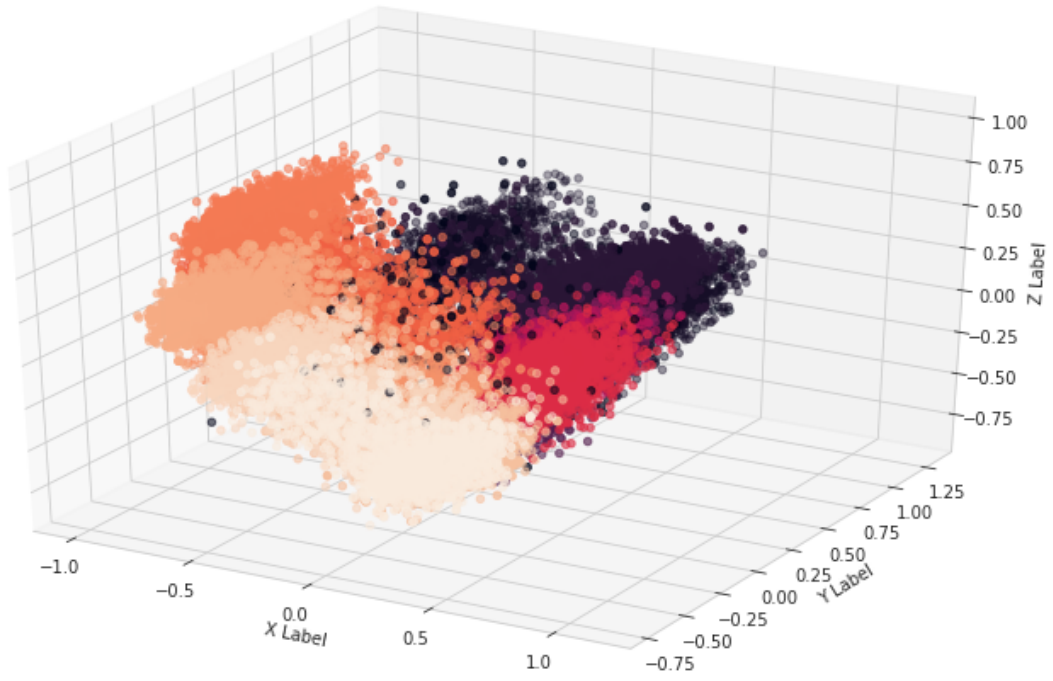
نمایش داده ها با این کلاستر بندی در 2 بعد به شکل بالا است و در 3 بعد هم به شکل زیراست.



حالت سوم (one hot encoding , dbscan):

برای dbscan 2 تا هایپر پارامتر داریم که یکی eps است و دیگری min samples.
که eps 0.5 گذاشتم min sample 5 که تنها فرقی که با قبلیا کرد این بود که dbscan داده های نویز را هم تشخیص داد و نتایج زیر هم نمایش داده ها در 2 و 3 بعد بود.

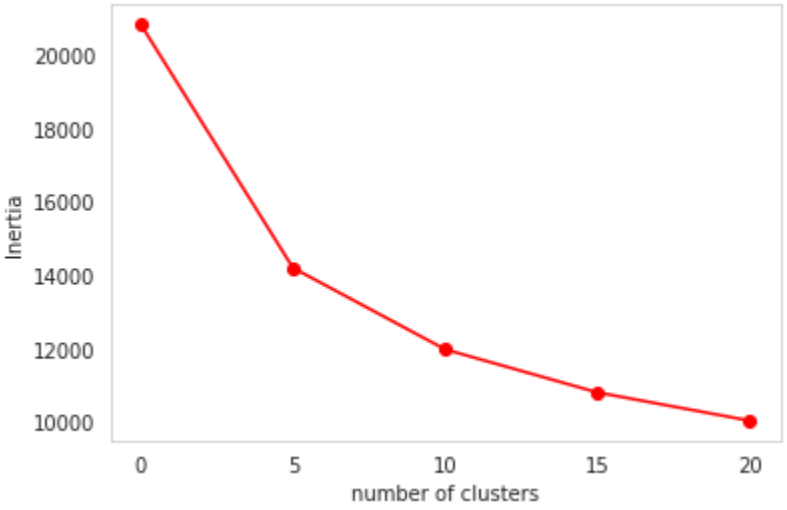


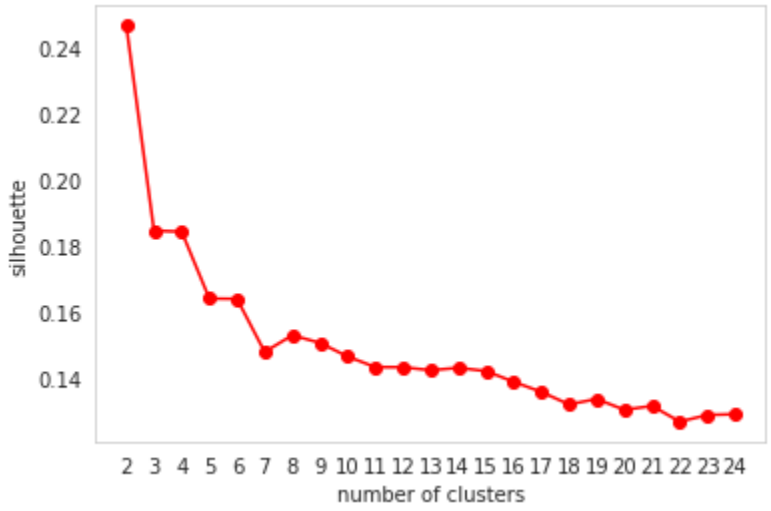


بعد از این نتایج متوجه می شویم که پس one hot کردن genre اصلا نمیتواند مناسب باشد.

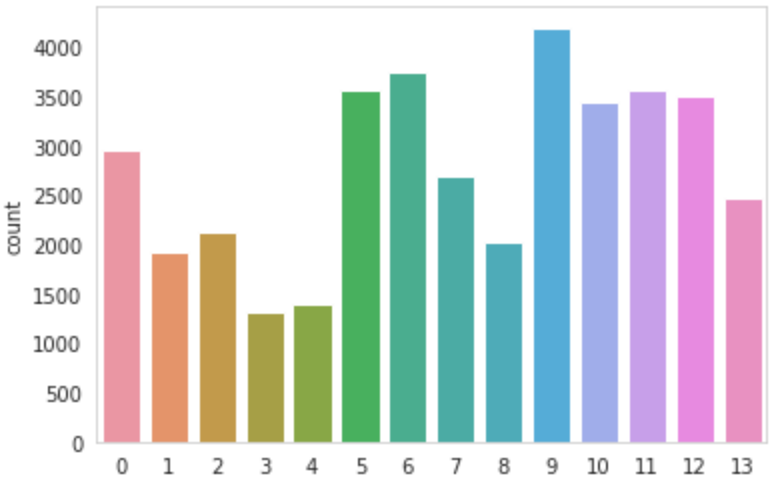
پس برای این بخش داده های ژانر را label encode کردم که به هر نوع داده یک عدد نسبت میدهد از 0 تا تعداد نوع های مختلف. و همچنین دوباره داده ها را minmax scale کردم.

حالت چهارم (Label Encoding , Kmeans):
دوباره معیار های inertia , silhouette را بررسی کردم که به نمودارهای آن ها به شکل زیر شد

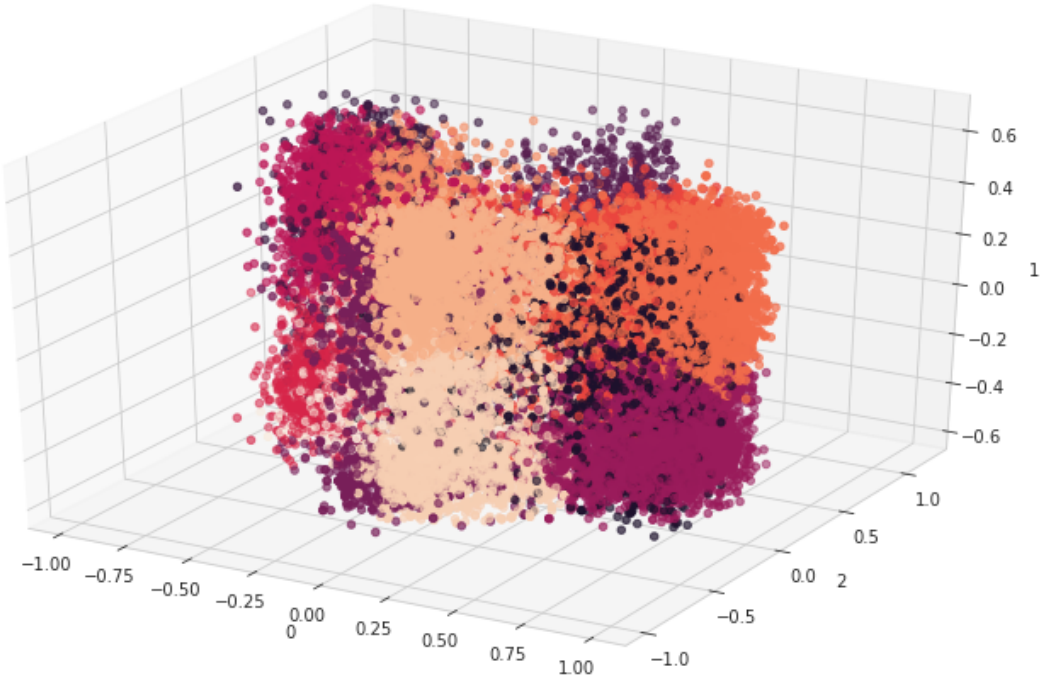


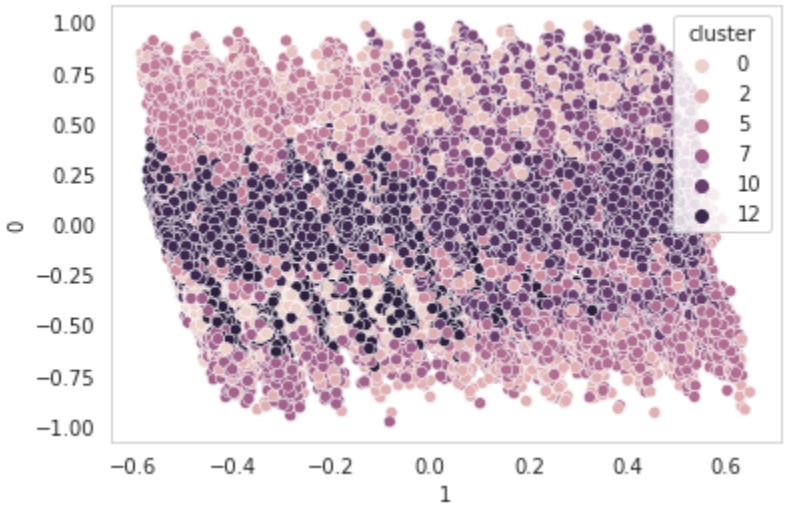


در نهایت 14 کلاستر رو به عنوان مدل نهایی امتحان کردم که تعداد داده های هر کلاستر به شکل زیر شد



بعد از کلاستر بندی بررسی کردم در هر کلاستر داده هایی که وجود دارند از چه ژانر های مختلفی اند که فهمیدم در هر کلاستر ژانر های متعددی وجود دارد که نشان میدهد label encode کردن ژانر بهتر است نسبت به one hot کردن.
نتایج زیر هم نمایش داده ها در 2 و 3 بعد است.



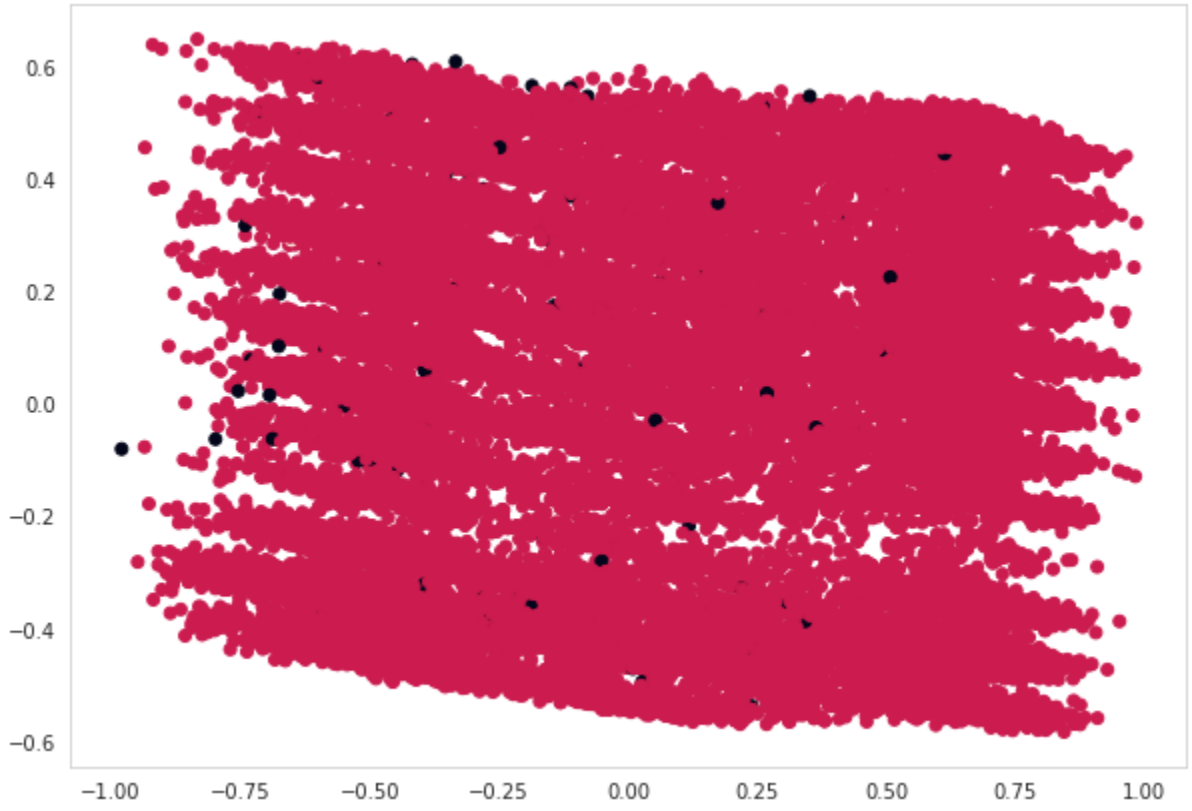


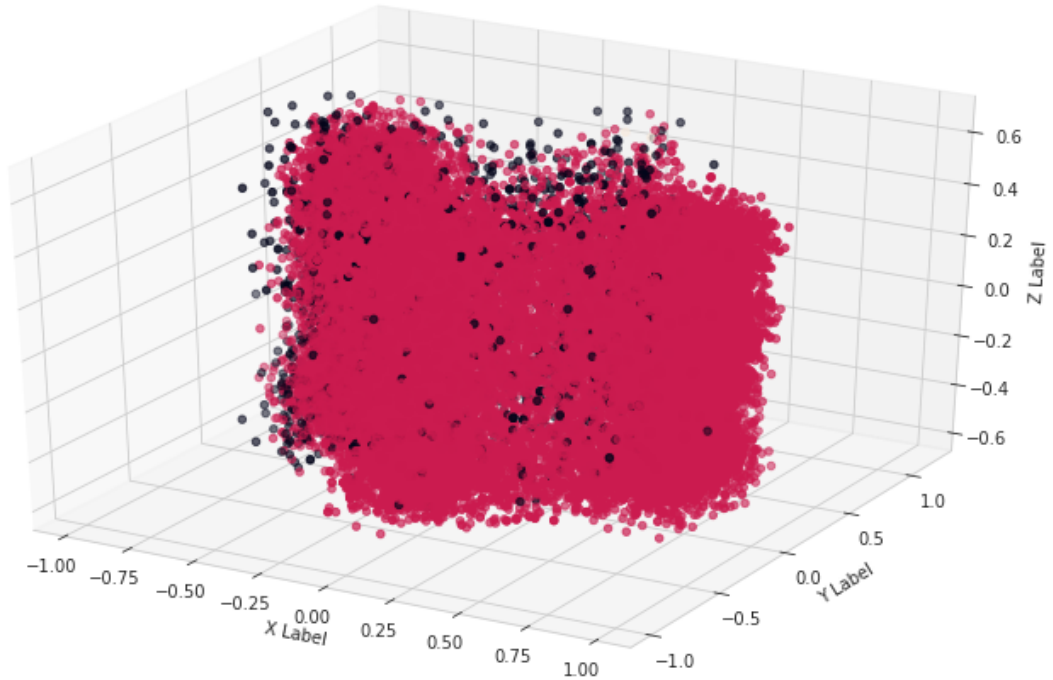
حالت پنجم (Label Encoding , hierarchical clustering):

برای این کلاستر بندی هم بهترین حالت 2 کلاستر کردن داده ها بود که اصلا به در ما در این مسئله نمیخورد.

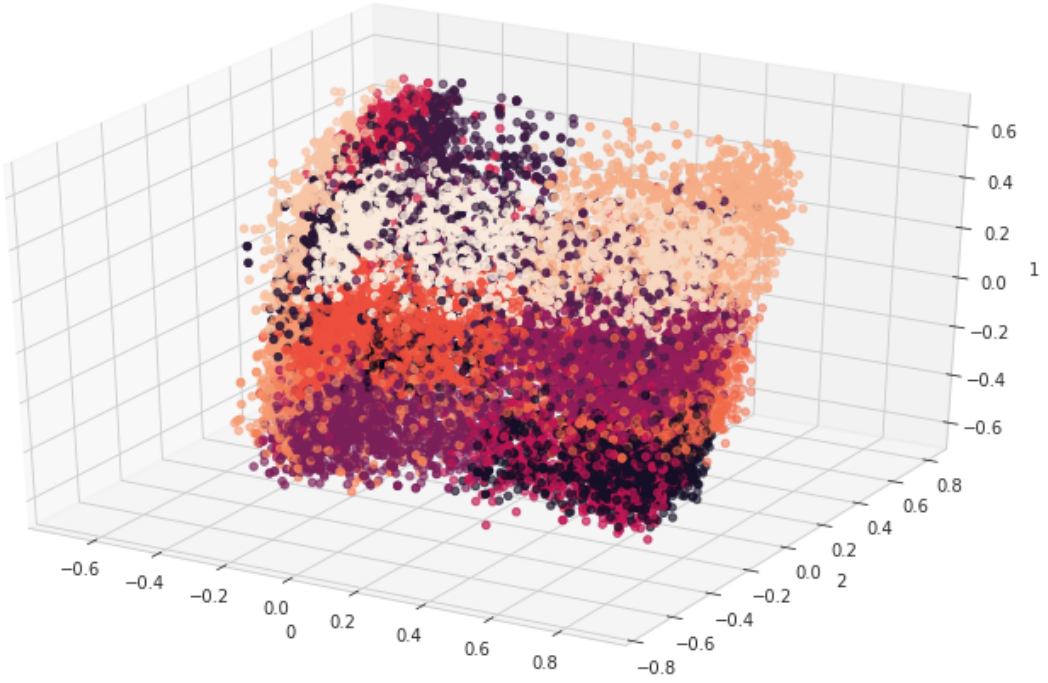
حالت ششم (Label Encoding , dbscan):

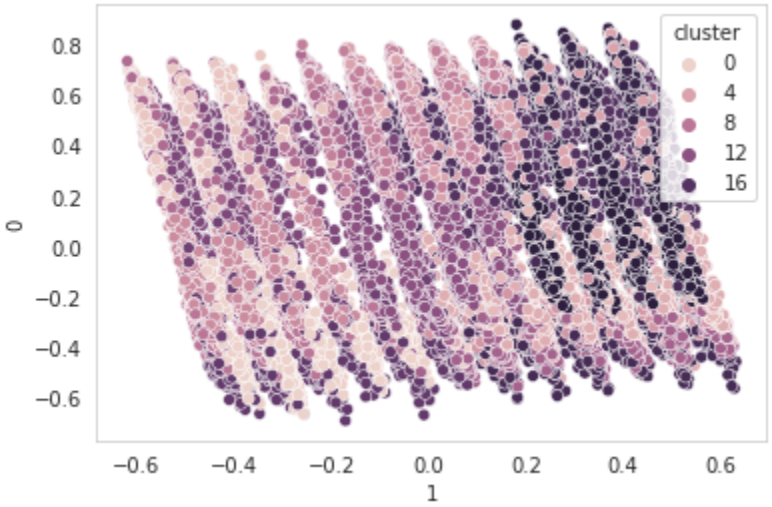
و اما مدل dbscan هم رو پارامتری های مختلف نتایج خوبی را نداشت به طور مثال یکی از مدل ها با پارامتر eps 0.4 و min_samples 10 به شکل زیر شد در 2 و 3 بعد:





اما در نهایت هیچکدام از این مدل ها را نمی توانیم به عنوان مدل نهایی برای recommender system خود انتخاب کنیم زیرا در گرفتن داده های جدید و گرفتن ویژگی آنها اصلا ستون ژانر وجود ندارد در نتیجه برای آخرین مدل و مدل نهایی ستون ژانر را هم حذف کرده و مدل kmeans که ظاهرا مناسب ترین مدل است (زیرا میتوانیم تعداد کلاستر را به طور دستی خودمان بالا ببریم) روی 20 کلاستر اجرا و ذخیره کردیم و نتایج آن رو داده ها به صورت 2 و 3 بعدی به شکل های زیر شد:





در این بخش من خود آهنگهایی را که در پلی لیست اسپاتیفای تهیه کردم را هم کل داده ها اضافه کردم چون من داده ها را minmax scale میکنم ممکن بود داده های آهنگ های من از min داده های کل کمتر و یا از max داده های کل بیشتر باشد.

در بخش کد اصلی مربوط به recommender بعد از اینکه مدل ما تشخیص میدهد که آهنگای ورودی من جزو چه کلاستر هایی هستند از 5 تا از کلاستر هایی که بیشتر من گوش میدهم 5 آهنگ از هر کلاستر به طور تصادفی انتخاب میکند و 5 آهنگ هر کلاستر را در یک فایل csv ذخیره میکند که درکل میشود 5 فایل csv.
اما task 2 که مربوط به top songs است من ورودی های خودم را کلا شامل 36 آهنگ میشد به مدل دادم و اینکه عضو چه کلاستری هستند و از هر کلاستر چند بار تکرار داریم را پیدا کردم به میزان تعداد تکرار از هر کلاستر به همون تعداد آهنگ جدید به صورت تصادفی از آن کلاستر در فایل csv ذخیره کردم که درنهایت یعنی 36 آهنگ جدید به طور کلی داریم که به مخاطب به شکل mix playlist پیشنهاد شده است.