



پروژه نهایی درس یادگیری ماشین

اعضای گروه:

سارا رستمی ۸۱۰۱۰۰۳۵۵

کیهان رعیتی ۸۱۰۱۰۰۳۶۱

انوشه سعادت‌ی ۸۱۰۱۰۰۳۷۴

سهیل صدقی ۸۱۰۱۰۰۳۹۹

نام استاد:

دکتر اعرابی

دکتر ابوالقاسمی

پاییز ۱۴۰۰

۱	پیش پردازش
۱	معرفی مجموعه داده
۲	استخراج ویژگی
۲	مجموعه داده پیش پردازش شده
۳	بررسی همبستگی ویژگی ها
۴	طبقه بندی
۴	طبقه بند logistic regression
۵	طبقه بند k-nearest neighbor
۷	طبقه بند Support Vector Machine
۹	طبقه بند MultiLayer Perceptron
۱۱	Feature Selection & Dimensionality Reduction
۱۲	خوشه بندی
۱۲	الگوریتم خوشه بندی k-means
۱۸	الگوریتم خوشه بندی k-medoid
۲۳	الگوریتم خوشه بندی Hierarchical

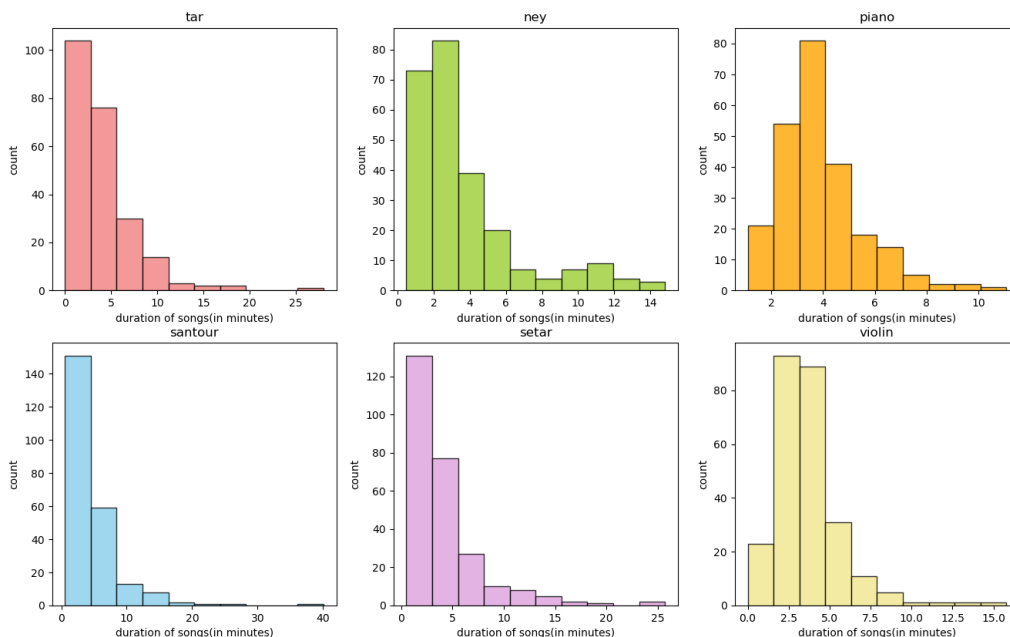
پیش‌پردازش

معرفی مجموعه داده

مجموعه داده‌ی اولیه که در اختیار ما قرار گرفت شامل تعدادی آهنگ در قالب فایل با فرمت mp3 بود. تعداد آهنگ‌های هر ساز به شرح زیر است: ۲۶۱ آهنگ برای ساز نی، ۲۴۶ آهنگ برای ساز پیانو، ۲۴۶ آهنگ برای ساز سنتور، ۲۷۰ آهنگ برای ساز سه‌تار، ۲۴۲ آهنگ برای ساز تار، ۲۶۳ آهنگ برای ساز ویولن.

مجموعه داده‌ی اولیه شامل تعدادی فایل خراب بود که در مرحله پیش‌پردازش آنها را شناسایی کردیم. تعداد فایل‌های خراب برای هر ساز به شرح زیر است: ۱۳ فایل خراب در فولدر ساز نی، ۸ فایل خراب در فولدر ساز پیانو، ۱۳ فایل خراب در فولدر ساز سنتور، ۹ فایل خراب در فولدر ساز سه‌تار، ۱۴ فایل خراب در فولدر ساز تار، ۵ فایل خراب در فولدر ساز ویولن.

هیستوگرام طول آهنگ‌ها را برای هر ساز در شکل ۱ مشاهده می‌کنید. میانگین طول آهنگ‌های هر یک از سازها عبارتند از: ۳.۶۷ دقیقه برای ساز نی، ۳.۸۹ دقیقه برای ساز پیانو، ۴.۶۶ دقیقه برای ساز سنتور، ۴.۱۸ دقیقه برای ساز سه‌تار، ۴.۱۳ دقیقه برای ساز تار، ۳.۶۳ دقیقه برای ساز ویولن.



شکل ۱- مدت زمان آهنگ‌ها به تفکیک ساز

استخراج ویژگی

در این مرحله با توجه به نوع داده‌ها (یعنی فایل صوتی) تعدادی ویژگی استخراج کردیم. برای استخراج این ویژگی‌ها، از کتابخانه‌ی librosa استفاده کردیم. ویژگی‌های استخراج شده‌ی ما، ویژگی‌های mfcc و zero-crossing هستند.

مجموعه داده پیش پردازش شده

در این مرحله مجموعه داده پیش پردازش شده را با استفاده از ویژگی‌های بدست آمده از مرحله قبل ساختیم. برای ساخت دیتاست، ۳۰ ثانیه‌ی اول هر آهنگ را در نظر گرفتیم. در واقع هر نمونه ما در دیتاست نشان دهنده‌ی مقادیر ویژگی‌های mfcc و zero crossing برای ۳۰ ثانیه‌ی اول یک فایل mp3 است. Label هر یک از نمونه‌ها، سازی است که آهنگ با آن نواخته شده است. چرا که هدف ما در این پروژه پیش‌بینی نوع سازها می‌باشد.

ویژگی MFCC (Mel-Frequency Cepstral Coefficients)

محاسبه ضرایب MFCC بر مبنای سیستم شنوایی انسان برای یک سیگنال صوتی انجام می‌شود. هر فریم سیگنال ابتدا در پنجره همینگ ضرب می‌شود و سپس از نتیجه تبدیل فوریه گسسته گرفته می‌شود. اندازه تبدیل فوریه گرفته شده محاسبه می‌شود و بر روی پوش طیف بدست آمده مراحل زیر برای استخراج ضرایب MFCC سیگنال انجام می‌شود.

فیلترها به شکل مثلثی بوده و شروع هر فیلتر از فرکانس مرکزی فیلتر قبلی و خاتمه آن در فرکانس مرکزی فیلتر بعدی است و ماکزیمم آن در فرکانس مرکزی خودش می‌باشد و CF فرکانس مرکزی فیلتر است. به طور کلی در کاربردهای تشخیص صوت تعداد ۱۲ تا ۲۰ ضریب MF پیشنهاد می‌شود که ما در این پروژه از ۲۰ ضریب mfcc استفاده کردیم.

مراحل استخراج ضرایب MFCC

۱. عبور سیگنال از فیلتر پیش تاکید
۲. فریم بندی سیگنال ورودی
۳. ضرب هر فریم در پنجره همینگ
۴. گرفتن تبدیل فوریه سریع از فریم‌های مرحله قبل
۵. انتقال فرکانس به حوزه mel
۶. ضرب خروجی بانک فیلتر در اندازه تبدیل فوریه سیگنال گرفتن تبدیل گسسته کسینوسی

ویژگی zero crossing

میزان عبور صفر میزان تغییرات علامت در امتداد سیگنال صدا است. تغییر علامت، یعنی سرعت تغییر سیگنال از مثبت به منفی یا از منفی به مثبت. از این ویژگی در تشخیص گفتار و بازیابی اطلاعات موسیقی استفاده می‌شود و یک ویژگی کلیدی برای طبقه بندی اصوات کوبه ای است.

ZCR به طور رسمی به این صورت تعریف می‌شود:

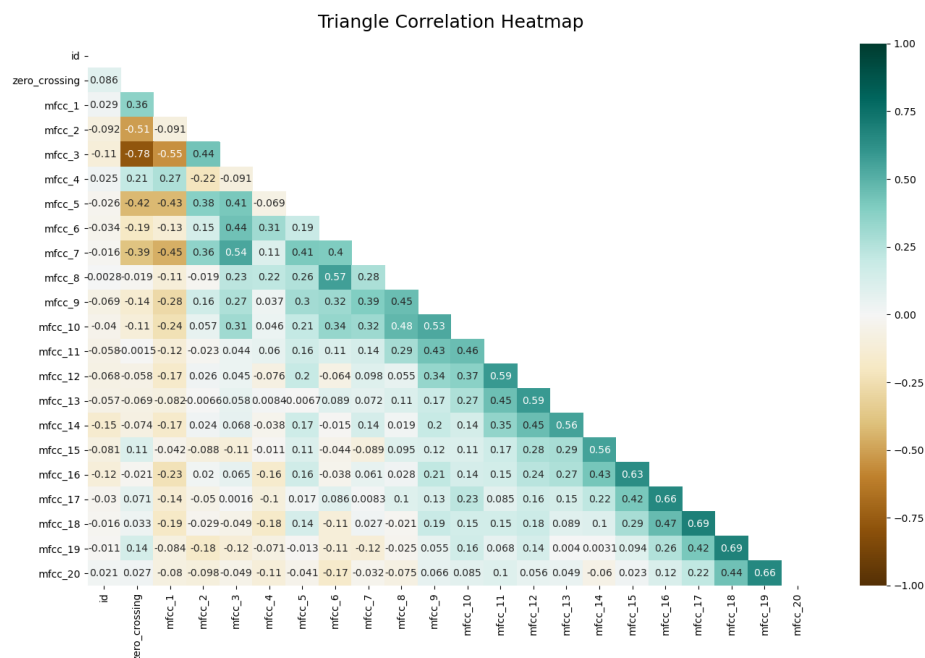
$$ZCR = \frac{1}{2N} \sum_{n=1}^N |\text{sign}(x[n]) - \text{sign}(x[n-1])|$$

پس دیتاست پیش‌پردازش شده حاوی ۲۳ ستون است. ستون اول نشان‌دهنده‌ی شماره‌ی آهنگ، ستون دوم نشان‌دهنده‌ی مقدار ویژگی zero-crossing برای ۳۰ ثانیه اول آهنگ، ستون‌های سوم تا بیست و دوم نشان‌دهنده‌ی مقادیر ویژگی‌های mfcc_1 تا mfcc_20 برای ۳۰ ثانیه اول آهنگ و ستون آخر که ستون label می‌باشد، نشان‌دهنده‌ی نام سازی است که آهنگ با آن نواخته شده‌است.

دیتاست پیش‌پردازش شده شامل ۱۴۹۰ سطر (در واقع همان نمونه) می‌باشد. که ۲۳۲ سطر آن مربوط به ساز تار، ۲۵۶ سطر آن مربوط به ساز نی، ۲۴۰ سطر آن مربوط به ساز پیانو، ۲۴۱ سطر آن مربوط به ساز سنتور، ۲۶۳ سطر آن مربوط به ساز سه‌تار و ۲۵۸ سطر آن مربوط به ساز ویولن می‌باشد.

بررسی همبستگی ویژگی‌ها

برای رسم ماتریس همبستگی از Spearman Correlation استفاده کردیم. چرا که با انجام normal hypothesis test بر روی هر یک از ویژگی‌ها دیدیم که تنها ویژگی‌های mfcc_1, mfcc_2, mfcc_4, mfcc_5, mfcc_6, mfcc_7 توزیع تقریباً نرمال دارند و بقیه از توزیع نرمال پیروی نمی‌کنند بنابراین Pearson Correlation قابل استفاده نبود. با رسم ماتریس correlation ویژگی‌ها می‌بینیم که بیشترین میزان همبستگی بین دو ویژگی حدود 0.78 می‌باشد. پس فقط با در نظر گرفتن correlation ، ویژگی‌ای را نمی‌توان به علت همبستگی زیاد با ویژگی دیگر حذف کرد. چرا که اغلب مقدار همبستگی 0.8 به بالا را به عنوان مقدار پایه برای حذف یکی از این ویژگی‌ها در نظر می‌گیرند(اینطور به نظر می‌رسد که این ویژگی اطلاعات چندانی نسبت به ویژگی‌ای که با آن همبستگی بالایی دارد به ما نمی‌دهد و با حذف آن سود بیشتری می‌بریم. به خاطر مشکل curse of dimensionality).



شکل ۲- ماتریس correlation ویژگی‌ها

طبقه‌بندی

در این گام برای بررسی عملکرد الگوریتم‌های دسته‌بندی مختلف بر روی دیتاست خود، داده‌ها را به سه دسته‌ی `train`، `test` و `validation` تقسیم کردیم. برای اینکار ابتدا داده‌ها را به دو دسته‌ی `train` و `test` تقسیم کردیم به طوریکه تعداد داده‌های `train` بیشتر باشد و سپس داده‌های `train` را به دو دسته‌ی `train` و `validation` تقسیم کردیم باز هم به طوریکه تعداد داده‌های `train` بیشتر باشد. از داده‌های `train` برای آموزش طبقه‌بندها استفاده کردیم. از داده‌های `validation` برای تنظیم پارامترهای الگوریتم خود استفاده کردیم و در نهایت عملکرد الگوریتم‌ها را بر روی داده‌های تست (که `unseen data` هستند) سنجیدیم. الگوریتم‌های مورد استفاده در این پروژه عبارتند از: `k-nearest neighbor`، `logistic regression`، `SVM` و `MLP`.

برای پیاده‌سازی این `classifier`ها از روش `one vs. all` استفاده کردیم. ما همچنین روش `one vs. one` را با استفاده از طبقه‌بندهای زیر سنجیدیم. ولی عملکرد این روش بدتر از روش `one vs. all` بود. از این رو در پیاده‌سازی نهایی از روش `one vs. all` استفاده کردیم. همچنین در ساخت مدل با استفاده از طبقه‌بند ها و همچنین در تقسیم داده‌ها به مجموعه‌های `train` و `test` و `validation`، `random_state = 42` قرار دادیم. تا نتایج ما، `reproducible` باشند.

طبقه‌بند `logistic regression`

این طبقه‌بند در کتابخانه‌ی `sklearn` موجود و قابل استفاده است. از آنجایی که در این طبقه‌بند از پارامتر خاصی استفاده نمی‌شود، نیاز به تنظیم پارامتر با استفاده از داده‌ی `validation` ندارد. در نتیجه با استفاده‌ی از داده‌ی `train` این طبقه‌بند آموزش دیده شد و عملکرد آن بر روی داده‌ی `test` سنجیده شد که نتیجه‌ی آن را در شکل ۲ مشاهده می‌کنید. در خط آخر خروجی عملکرد این طبقه‌بند که در شکل ۳ قابل مشاهده است، میزان دقت (`accuracy`) طبقه‌بند را برای تشخیص هر یک از کلاس‌ها (سازها) می‌بینید.

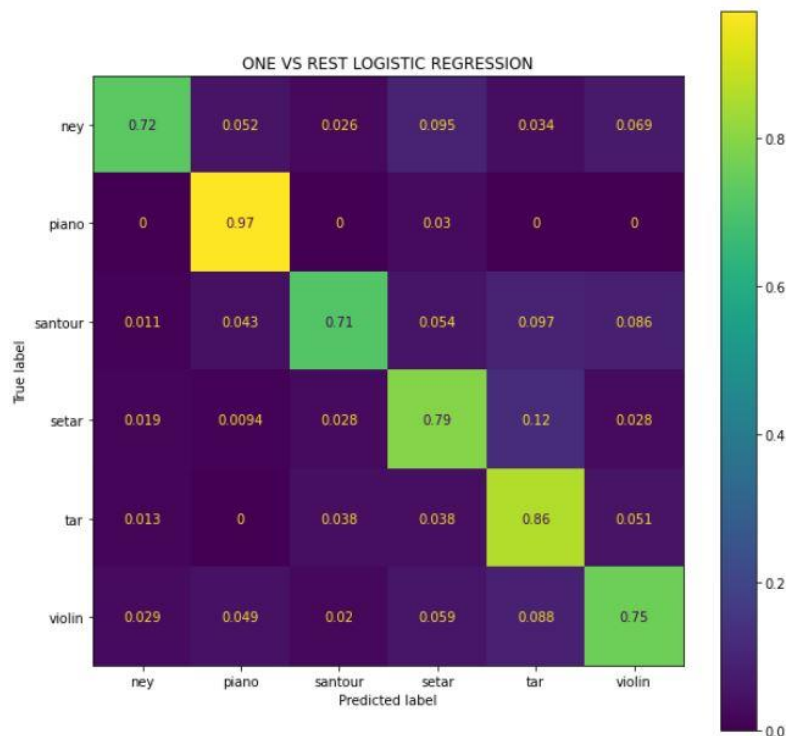
	precision	recall	f1-score	support
ney	0.92	0.72	0.81	116
piano	0.86	0.97	0.91	101
santour	0.86	0.71	0.78	93
setar	0.75	0.79	0.77	106
tar	0.66	0.86	0.74	78
violin	0.77	0.75	0.76	102
accuracy			0.80	596
macro avg	0.80	0.80	0.80	596
weighted avg	0.81	0.80	0.80	596

```
['ney' 'piano' 'santour' 'setar' 'tar' 'violin']  
[0.72413793 0.97029703 0.70967742 0.79245283 0.85897436 0.75490196]
```

شکل ۳- عملکرد طبقه‌بند `logistic regression`

با توجه به شکل ۳ می‌توان مشاهده کرد که این طبقه‌بند در تشخیص ساز پیانو بسیار موفق عمل می‌کند.

ماتریس آشفتگی این طبقه‌بند در شکل ۴ قابل مشاهده است.

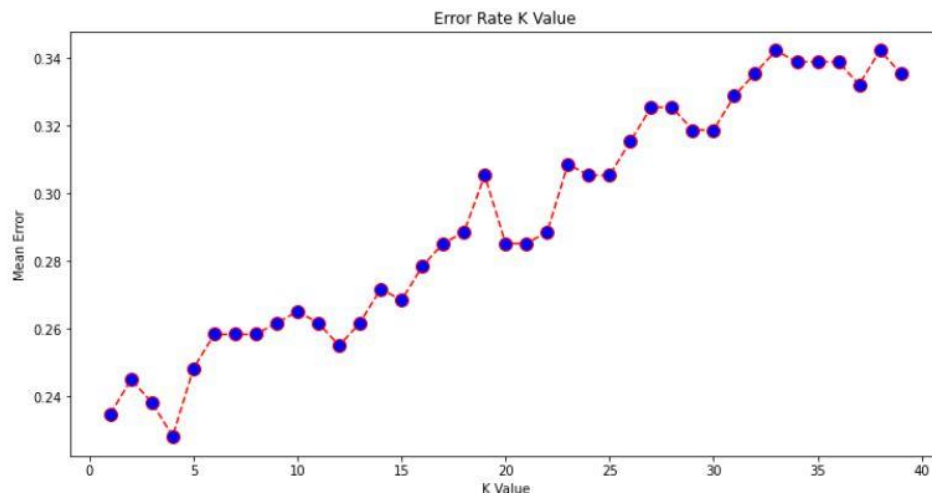


شکل ۴- ماتریس آشفته‌گی طبقه‌بند *logistic regression*

با توجه به شکل ۴ می‌توان دید که ساز پیانو به ندرت به عنوان یک ساز دیگر تشخیص داده شده (تنها در ۳٪ مواقع). علت آن هم می‌تواند ماهیت و نوای متفاوت ساز پیانو نسبت به سایر سازها باشد. همچنین می‌بینیم بیشترین تشخیص اشتباه دو ساز به جای هم مربوط به جفت ساز تار و سه‌تار است. در ۱۲٪ مواقع طبقه‌بند *logistic regression* ساز سه‌تار را اشتباهاً ساز تار می‌شمارد. علت این موضوع نیز شباهت زیاد این دو ساز از نظر نوع ساخت و نوا به هم می‌باشد.

طبقه‌بند k-nearest neighbor

این طبقه‌بند در کتابخانه‌ی *sklearn* موجود و قابل استفاده است. در این طبقه‌بند پارامتر k باید به گونه‌ای مقدار بگیرد که الگوریتم بهترین عملکرد را روی داده‌ی *validation* داشته باشد. پس ابتدا میانگین خطا برحسب مقدار k را با استفاده از داده‌ی *validation* رسم می‌کنیم. این نمودار در شکل ۵ قابل مشاهده است و میزان خطای طبقه‌بند را به ازای مقادیر $k=1$ تا $k=40$ نشان می‌دهد. با توجه به شکل ۵ می‌توان نتیجه گرفت که بهترین مقدار k برابر با ۴ می‌باشد. البته این مقدار دقیق با استفاده از کد بدست آورده شده است.



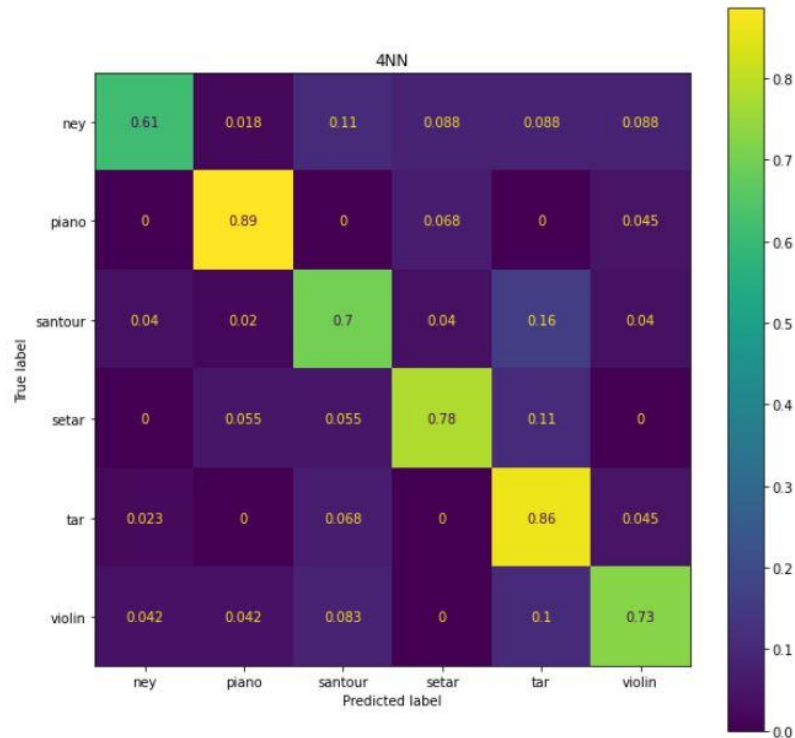
شکل ۵- نمودار میانگین خطای طبقه‌بند knn برحسب مقدار k

حال پس از بدست آوردن مقدار k ی مناسب ($k=4$)، طبقه‌بند را با داده‌ی **train** آموزش می‌دهیم و سپس عملکرد طبقه‌بند را روی داده‌ی **test** می‌سنجیم که می‌توان خروجی آن را در شکل ۶ مشاهده کرد.

	precision	recall	f1-score	support
ney	0.88	0.61	0.72	57
piano	0.85	0.89	0.87	44
santour	0.69	0.70	0.69	50
setar	0.81	0.78	0.80	55
tar	0.61	0.86	0.72	44
violin	0.76	0.73	0.74	48
accuracy			0.76	298
macro avg	0.77	0.76	0.76	298
weighted avg	0.77	0.76	0.76	298
['ney' 'piano' 'santour' 'setar' 'tar' 'violin']				
[0.61403509 0.88636364 0.7 0.78181818 0.86363636 0.72916667]				

شکل ۶- عملکرد طبقه‌بند knn با $k=4$

باتوجه به شکل ۶ می‌توان مشاهده کرد که این طبقه‌بند در تشخیص ساز پیانو و تار در مقایسه با سایر سازها بسیار موفق‌تر عمل می‌کند. این موضوع با دقت به ماتریس آشفتگی این طبقه‌بند که در شکل ۷ قابل مشاهده است، نیز مشهود است.



شکل ۷- ماتریس آشفته‌گی طبقه‌بند knn با $k=4$

با توجه به شکل ۷ می‌بینیم بیشترین تشخیص اشتباه دو ساز به جای هم مربوط به جفت ساز تار و سنتور است. در ۱۶٪ مواقع طبقه‌بند knn ساز سنتور را اشتباهاً ساز تار می‌شمارد. و پس از آن جفت ساز تار و سه‌تار (طبقه‌بند knn ساز سه‌تار را در ۱۱٪ مواقع اشتباهاً به عنوان ساز تار تشخیص می‌دهد).

طبقه‌بند Support Vector Machine

این طبقه‌بند در کتابخانه‌ی `sklearn` موجود و قابل استفاده است. طبقه‌بند `SVM` دارای پارامترهای `kernel`، `C` و `gamma` می‌باشد. برای بدست آوردن مقادیر مناسب این پارامترها که منجر به بهترین عملکرد طبقه‌بند روی داده‌ی `validation` شود، از `grid search` استفاده کردیم. `Grid search` برای سنجیدن عملکرد طبقه‌بند از معیار `accuracy` استفاده می‌کند. `search` به عنوان خروجی مقادیر `kernel = RBF` و `C = 1000` و `gamma = 0.001` را برای پارامترهای ذکر شده در نظر گرفت.

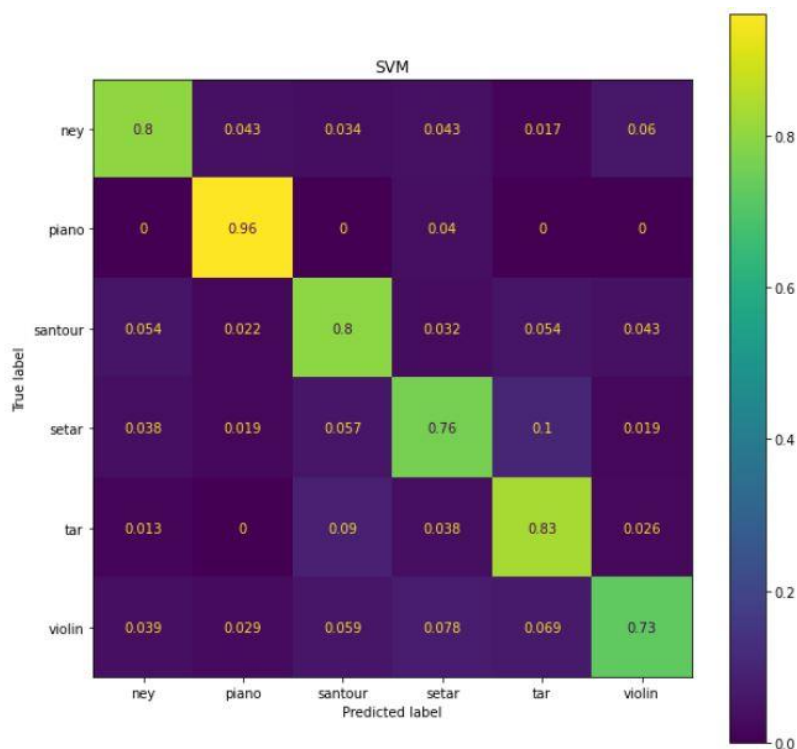
حال طبقه‌بند را با مقادیر بدست آمده برای پارامترها، با داده‌ی `train` آموزش می‌دهیم و سپس عملکرد آن را روی داده‌ی `test` می‌سنجیم که در شکل ۸ قابل مشاهده است.

	precision	recall	f1-score	support
ney	0.87	0.80	0.83	116
piano	0.89	0.96	0.92	101
santour	0.76	0.80	0.78	93
setar	0.78	0.76	0.77	106
tar	0.72	0.83	0.77	78
violin	0.83	0.73	0.77	102
accuracy			0.81	596
macro avg	0.81	0.81	0.81	596
weighted avg	0.81	0.81	0.81	596

```
['ney' 'piano' 'santour' 'setar' 'tar' 'violin']
[0.80172414 0.96039604 0.79569892 0.76415094 0.83333333 0.7254902 ]
```

شکل ۸- عملکرد طبقه‌بند SVM با پارامترهای $\gamma = 0.001$ و $C=1000$ و $\text{kernel}=RBF$

باتوجه به شکل ۸ می‌توان مشاهده کرد که این طبقه‌بند در تشخیص ساز پیانو در مقایسه با سایر سازها بسیار موفق‌تر عمل می‌کند. این موضوع با دقت به ماتریس آشفتگی این طبقه‌بند که در شکل ۹ قابل مشاهده است، نیز مشهود است.



شکل ۹- ماتریس آشفتگی طبقه‌بند SVM با پارامترهای $\gamma = 0.001$ و $C=1000$ و $\text{kernel}=RBF$

با توجه به شکل ۹ می‌بینیم بیشترین تشخیص اشتباه دو ساز به جای هم مربوط به جفت ساز تار و سه‌تار است. طبقه‌بند svm ساز سه‌تار را در ۱۰٪ مواقع اشتباهاً به عنوان ساز تار تشخیص می‌دهد.

طبقه‌بند MultiLayer Perceptron

این طبقه‌بند در کتابخانه‌ی sklearn موجود و قابل استفاده است. این طبقه‌بند هم دارای تعدادی پارامتر است که نیاز به تنظیم دارند. برای تنظیم پارامترها در این طبقه‌بند هم از Grid Search استفاده می‌کنیم. در اینجا هم Grid Search برای سنجیدن عملکرد طبقه‌بند از معیار accuracy استفاده می‌کند. با اجرای Grid Search بر روی داده‌ی validation مقادیر زیر برای پارامترهای MLP بدست می‌آید:

- activation: tanh
- alpha: 0.0001
- hidden_layer_sizes: (256, 128, 64, 32)
- learning_rate: constant
- max_iter: 100
- solver: adam

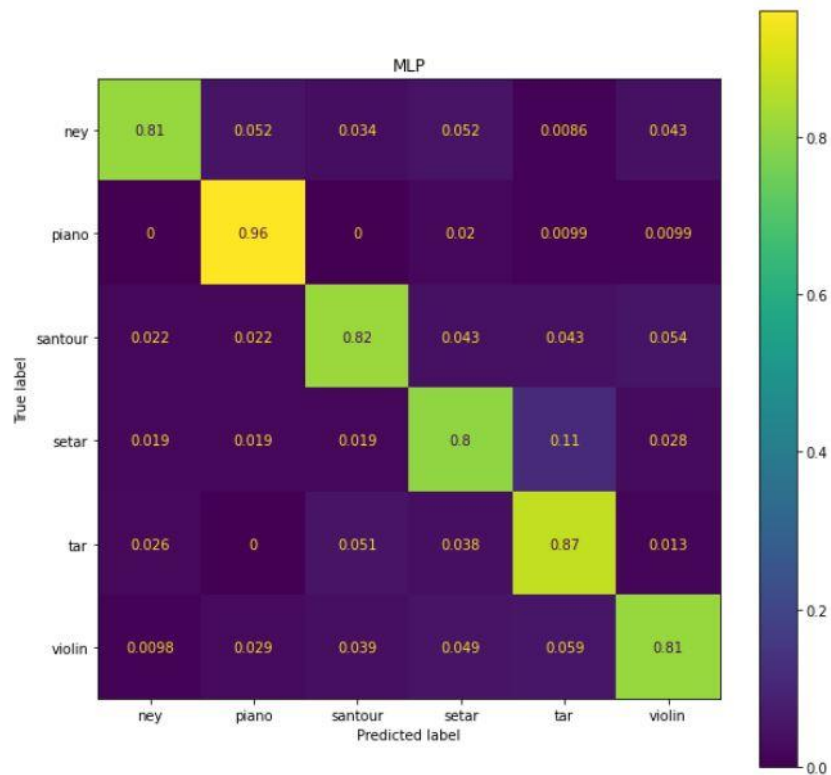
حال طبقه‌بند را با مقادیر بدست آمده برای پارامترها، با داده‌ی train آموزش می‌دهیم و سپس عملکرد آن را روی داده‌ی test می‌سنجیم که در شکل ۱۰ قابل مشاهده است.

	precision	recall	f1-score	support
ney	0.93	0.81	0.87	116
piano	0.88	0.96	0.92	101
santour	0.84	0.82	0.83	93
setar	0.81	0.80	0.81	106
tar	0.74	0.87	0.80	78
violin	0.85	0.81	0.83	102
accuracy			0.84	596
macro avg	0.84	0.85	0.84	596
weighted avg	0.85	0.84	0.84	596


```
['ney' 'piano' 'santour' 'setar' 'tar' 'violin']
[0.81034483 0.96039604 0.8172043 0.80188679 0.87179487 0.81372549]
```

شکل ۱۰ - عملکرد طبقه‌بند MLP

باتوجه به شکل ۱۰ می‌توان مشاهده کرد که این طبقه‌بند در تشخیص ساز پیانو در مقایسه با سایر سازها بسیار موفق‌تر عمل می‌کند. این موضوع با دقت به ماتریس آشفتگی این طبقه‌بند که در شکل ۱۱ قابل مشاهده است، نیز مشهود است.



شکل ۱-۱- ماتریس آشفتگی طبقه‌بند MLP

می‌بینیم که تشخیص اشتباه ساز تار به جای سه‌تار در طبقه‌بند MLP نیز بیشترین درصد را دارد (طبقه‌بند mlp ساز سه‌تار را در ۱۱٪ مواقع اشتباهش به عنوان ساز تار تشخیص می‌دهد).

در کل دیدیم که طبقه‌بندها در تمایز ساز تار و سه‌تار با دشواری بیشتری مواجه هستند. همانطور که توضیح داده شد این دو ساز از نظر ساخت و نوع نوا شباهت زیادی بهم دارند و در نتیجه ایجاد تمایز بین آنها دشوارتر می‌باشد. یک راه برای کمک به حل این موضوع می‌تواند اضافه کردن ویژگی (یا ویژگی‌های) discriminant ی باشد که بتواند تفاوت بین آنها را به خوبی capture کند. پیدا کردن و بررسی چنین ویژگی‌ای کار پیچیده و زمانبری است که ذیق وقت از بحث ما خارج است.

Feature Selection & Dimensionality Reduction

به منظور بهبود دقت و همچنین سرعت طبقه‌بندها می‌توان feature conditioning انجام داد که خود شامل feature selection و dimensionality reduction می‌باشد.

در feature selection زیرمجموعه‌ای از ویژگی‌های اولیه انتخاب می‌شوند. این ویژگی‌ها باید به گونه‌ای انتخاب شوند که discriminant و invariant باشند. برای feature selection می‌توان از الگوریتم‌های مختلفی استفاده کرد. ما برای این کار از الگوریتم Sequential Forward Selection استفاده کردیم. این الگوریتم از ۲۱ ویژگی اولیه، ۱۳ ویژگی را انتخاب کرد. ما با استفاده از این ۱۳ ویژگی الگوریتم‌های شرح داده شده در بالا را دوباره اجرا کردیم ولی عملکرد همه‌ی طبقه‌بندها با این ۱۳ ویژگی کاهش یافت.

در dimensionality reduction از ترکیب ویژگی‌های اولیه تعدادی ویژگی جدید تولید می‌شوند. ما برای این کار از روش‌های PCA و LDA استفاده کردیم. هر دوی این روش‌ها ترکیب خطی از ویژگی‌های اولیه را به ما می‌دهند. اما با اِپلای کردن این روش‌ها و اجرای مجدد مراحل train و test طبقه‌بندها، عملکرد طبقه‌بندها افت پیدا کرد.

یکی از دلایل اینکه روش‌های feature conditioning برای مسئله‌ی ما جواب ندادند می‌تواند این باشد که تعداد ویژگی‌های انتخاب شده‌ی ما خیلی زیاد نبود (تنها ۲۱ ویژگی داشتیم) و با حذف برخی از آنها یا ترکیب آنها بخشی از اطلاعاتی که این ویژگی‌ها در مورد نمونه‌ها به ما می‌دهند از دست می‌رود. همچنین همانطور که در قسمت استخراج ویژگی هم نشان دادیم، این ویژگی‌ها میزان correlation خیلی زیادی باهم ندارند و در نتیجه با حذف هر یک از آنها ممکن بخشی از اطلاعات ارزشمند برای شناخت نمونه‌ها از دست برود. از این رو ما در پروژه ویژگی‌ای را حذف نکردیم و همچنین تعداد بعد فضای ویژگی را کاهش ندادیم و با کل ۲۱ ویژگی کار کردیم.

خوشه‌بندی

خوشه‌بندی روشی بدون ناظر است که در آن نمونه‌های مجموعه داده بر اساس معیار شباهت و یا تفاوت به خوشه‌هایی تقسیم می‌شوند. در این پروژه برای خوشه‌بندی از الگوریتم‌های k-means و k-medoid و hierarchical استفاده کردیم که همگی روش‌های مبتنی بر فاصله هستند که در تمام آن‌ها برای محاسبه فاصله از فاصله اقلیدسی استفاده کردیم. سپس عملکرد این الگوریتم‌های خوشه‌بندی را به ازای تعداد خوشه‌های ۱ تا ۶ ارزیابی کردیم. در این پیاده‌سازی‌ها مقدار random_state برابر ۰ در نظر گرفته شده است.

الگوریتم خوشه‌بندی k-means

این الگوریتم به عنوان ورودی تعداد k خوشه‌ها را دریافت می‌کند. سپس به صورت تصادفی k نمونه را به مرکز اولیه خوشه‌ها در نظر می‌گیرد. سپس هر نمونه موجود در مجموعه داده را به خوشه‌ای که کمترین فاصله را با مرکز آن دارد انتساب می‌دهد. سپس مرکز خوشه‌ها بروز رسانی می‌شود. این فرآیند برای تعداد از پیش تعیین شده‌ای تکرار انجام می‌شود.

پارامتر init به صورت پیش فرض برابر k-means++ است. پس پیاده‌سازی روش K-means به این نتایج رسیدیم:

K=۱

purity 1 : 0.17651006711409395

با $k=1$ همه‌ی داده‌ها در یک کلاستر قرار داده است. با $k=1$ مقدار purity برابر ۰.۱۷ شده است که به این معناست که ۱۷ درصد داده‌ها دارای لیبل‌هایی هستند که بیشترین تعداد را نسبت به بقیه دارد.

با $K=1$ ، تعداد نمونه‌های سازهای نی – پیانو – سنتور – سه‌تار – تار – ویولن در تک خوشه به ترتیب در ماتریس زیر آمده است.

['ney', 'piano', 'santour', 'setar', 'tar', 'violin']

[[256], [240], [241], [263], [232], [258]]

با توجه به این ماتریس بیشترین تعداد نمونه در این تک خوشه مربوط به ساز سه‌تار با مقدار ۲۶۳ می‌باشد. البته از آنجایی که در مجموعه داده پیش‌پردازش شده تعداد نمونه‌های ساز سه‌تار بیشتر است این اتفاق رخ داده است.

K=۲

purity 2 : 0.25100671140939596

با $K=2$ ، تعداد نمونه‌های سازهای نی - پیانو - سنتور - سه تار - تار - ویولن در دو خوشه به ترتیب در ماتریس زیر آمده است.

['ney', 'piano', 'santour', 'setar', 'tar', 'violin']

[[99 157], [4 236], [138 103], [74 189], [19 213], [90 168]]

بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی اول ۱۳۸ تاست که به ساز سنتور تعلق دارد. از این رو برچسب خوشه‌ی اول سنتور می‌شود. و بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی دوم ۲۳۸ تا بوده که برای ساز پیانوست. از این رو برچسب خوشه‌ی دوم پیانو می‌شود.

در این قسمت سنتور label ۱ است و کلاستر ۱ هم تعداد بیشتری سنتور دارد و این نشان می‌دهد که از داده‌های کلاستر ۲ به حد خوبی فاصله داشته است. پیانو بیشتر داده‌های آن در لیبل ۲ قرار دارد و کلاستر ۲ هم تعداد بیشتری پیانو دارد و به این معناست که داده‌های آن به کلاستر ۱ شباهت خیلی کمی دارد که داده‌های سنتور در آن قرار دارد. این مورد به صورت خوبی برای تار هم برقرار است.

با $k=2$ ، به طور میانگین ۲۵ درصد داده‌های هر کلاستر لیبل کلاستر را دارد.

$K=3$:

purity 3 : 0.2906040268456376

با $K=3$ ، نی - پیانو - سنتور - سه تار - تار - ویولن به ترتیب در ماتریس زیر آمده است.

['ney', 'piano', 'santour', 'setar', 'tar', 'violin']

[[36 119 101], [0 9 231], [68 134 39], [32 103 128], [1 79 152], [28 131 99]]

بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی اول ۶۸ تاست که به ساز سنتور تعلق دارد. از این رو برچسب خوشه‌ی اول سنتور می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی دوم ۱۳۴ تا بوده که آن نیز برای ساز سنتور است. از این رو برچسب خوشه‌ی دوم نیز سنتور می‌شود. و بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی سوم ۲۳۱ تا بوده که برای ساز پیانوست. از این رو برچسب خوشه‌ی سوم پیانو می‌شود.

بیشتر داده های سنتور در کلاستر ۲ است ولی نسبت به بقیه داده ها در کلاستر ۱-۲ بیشتر است، در این صورت می توان گفت که سنتور نسبت به کلاستر ۳ خیلی دور است و به کلاستر ۲ نزدیک تر از کلاستر ۱ است. بیشتر پیانو در کلاستر ۳ است و کلاستر ۳ هم دارای تعداد بیشتری پیانو نسبت به بقیه است. با توجه به ماتریس بالا می توان گفت که کلاستر ۳ از کلاستر ۱ به نسبت خوبی فاصله دارد ولی در بیشتر داده ها کلاستر ۲-۳ به خوبی جدا نشده اند.

با $k=3$ تا ۱۷ تکرار به جواب رسیده است و به طور میانگین ۲۹ درصد داده های هر کلاستر لیبل کلاستر را دارد.

$K=4$:

purity 4 : 0.31543624161073824

با $K=4$ ، نی - پیانو - سنتور - سه تار - تار - ویولن به ترتیب در ماتریس زیر آمده است.

['ney', 'piano', 'santour', 'setar', 'tar', 'violin']

[[95 16 71 74][12 0 4 224][94 32 93 22][115 15 51 82][121 0 15 96]

[114 7 75 62]]

بیشترین تعداد داده ی مربوط به یک ساز در خوشه ی اول ۱۲۱ تاست که به ساز تار تعلق دارد. از این رو برچسب خوشه ی اول تار می شود. بیشترین تعداد داده ی مربوط به یک ساز در خوشه ی دوم ۳۲ تا بوده که برای ساز سنتور است. از این رو برچسب خوشه ی دوم سنتور می شود. بیشترین تعداد داده ی مربوط به یک ساز در خوشه ی سوم ۹۳ تا بوده که برای ساز سنتور است. از این رو برچسب خوشه ی سوم نیز سنتور می شود و بیشترین تعداد داده ی مربوط به یک ساز در خوشه ی چهارم ۲۲۴ تا بوده که برای ساز پیانو است. از این رو برچسب خوشه ی چهارم پیانو می شود.

پیانو به طور خوبی از کلاستر ۱-۲-۳ فاصله دارد و بیشتر پیانو در کلاستر ۴ قرار دارد و کلاستر ۴ نسبت به بقیه داده ها دارای پیانو بیشتری است. سنتور در کلاستر ۱-۳ بیشتر است ولی کلاستر ۲-۳ دارای سنتور بیشتری نسبت به بقیه است که می توان گفت به صورت خوبی جدا نشده اند. کلاستر ۱ که دارای label تار است با توجه به مقادیر ماتریس بالا در رابطه با تار می توان گفت از کلاستر ۲-۳ به اندازه خوبی فاصله دارد.

با $k=4$ تا ۲۶ تکرار به جواب رسیده است و به طور میانگین ۳۱ درصد داده های هر کلاستر لیبل کلاستر را دارد.

$K=5$:

purity 5 : 0.34697986577181206

با $K=5$ ، نی - پیانو - سنتور - سه تار - تار - ویولن به ترتیب در ماتریس زیر آمده است.

['ney', 'piano', 'santour', 'setar', 'tar', 'violin']

[[54 84 29 80 9], [212 5 0 23 0], [12 89 62 72 6], [54 58 29 119 3],

[47 35 1 149 0], [43 85 28 101 1]]

بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی اول ۲۱۲ تاست که به ساز پیانو تعلق دارد. از این رو برچسب خوشه‌ی اول پیانو می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی دوم ۸۹ تا بوده که برای ساز سنتور است. از این رو برچسب خوشه‌ی دوم سنتور می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی سوم ۶۲ تا بوده که برای ساز سنتور است. از این رو برچسب خوشه‌ی سوم نیز سنتور می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی چهارم ۱۴۹ تا بوده که برای ساز تار است. از این رو برچسب خوشه‌ی چهارم تار می‌شود و بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی پنجم ۹ تا بوده که برای ساز نی است. از این رو برچسب خوشه‌ی پنجم نی می‌شود.

در این قسمت به صورت بهتری کلاستر بندی شده است. به صورتی که کلاستر ۱ دارای تعداد بیشتری پیانو است و تعداد بیشتر پیانو در کلاستر ۱ قرار گرفته است و از کلاستر های دیگر به حد خوبی فاصله دارد. کلاستر ۲-۳ سنتور است که ماتریس بالا نشان می‌دهد به صورت خوبی جدا نشده اند. کلاستر ۴ دارای تعداد بیشتر تار است ولی با توجه به ماتریس بالا صدای تار و سه تار و ویولن به هم خیلی شبیه در نظر گرفته شده است و نتوانسته به خوبی آن ها را از هم جدا کند. کلاستر ۵ دارای تعداد بیشتر نی است که اصلا رضایت بخش نیست چون داده های نی به کلاستر ۲ بیشتر از ۴ شبیه است. به خوبی نتوانسته آن ها را از هم جدا کند.

با $k=5$ تا ۱۶ تکرار به جواب رسیده است و به طور میانگین ۳۴ درصد داده های هر کلاستر لیبل کلاستر را دارد.

K=6

purity 6 : 0.3530201342281879

با $K=6$ ، نی - پیانو - سنتور - سه تار - تار - ویولن به ترتیب در ماتریس زیر آمده است.

['ney', 'piano', 'santour', 'setar', 'tar', 'violin']

[80 52 129 70 186 9]

[[68 23 67 57 32 9], [6 0 44 4 186 0], [78 52 31 70 4 6], [70 25 101 39 25 3], [73 0 129 15 15 0], [80 18 72 65 22 1]]

بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی اول ۸۰ تاست که به ساز ویولن تعلق دارد. از این رو برچسب خوشه‌ی اول ویولن می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی دوم ۵۲ تا بوده که برای ساز سنتور است. از این رو برچسب خوشه‌ی دوم سنتور می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی سوم ۱۲۹ تا بوده که برای ساز تار است. از این رو برچسب خوشه‌ی سوم تار می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی چهارم ۷۰ تا بوده که برای ساز سنتور است. از این رو برچسب خوشه‌ی چهارم نیز سنتور می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی پنجم ۱۸۶ تا بوده که برای ساز پیانو است. از این رو برچسب خوشه‌ی پنجم پیانو می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی ششم ۹ تا بوده که برای ساز نی است. از این رو برچسب خوشه‌ی ششم نی می‌شود.

نتایج $k=5$ و $k=6$ هردو به نسبت بهتر از مقادیر دیگر برای k است. بیشتر داده‌های ویولن در کلاستر ۱ است و کلاستر ۱ هم دارای بیشتر ویولن است و نسبت به ماتریس بالا پیانو به کلاستر ۳-۴ شبیه تر است و از کلاسترهای ۲-۵-۶ به شدت دور است.

بیشتر داده‌های سنتور در کلاستر ۱ و ۳ است ولی کلاستر ۲ و ۴ دارای تعداد بیشتر سنتور هستند این به معنی جدا پذیری غیر دقیق است.

بیشتر داده‌های تار در کلاستر ۳ قرار دارد و کلاستر ۳ هم دارای بیشتر تار است و نسبت به ماتریس بالا از کلاسترهای ۲-۴-۵-۶ به شدت دور است و به داده‌های ۱ نزدیک تر از بقیه است. بیشتر داده‌های پیانو در کلاستر ۵ قرار دارد و کلاستر ۵ هم دارای بیشتر پیانو است و نسبت به ماتریس بالا از کلاسترهای ۱-۲-۳-۴-۶ به شدت دور است و کلاسترینگ خوبی برای خوشه بندی آن انجام شده است. داده‌های نی بیشتر در کلاستر ۱ قرار دارد ولی کلاستر ۶ دارای تعداد بیشتر داده نی است که دقت خیلی پایینی ارائه داده است.

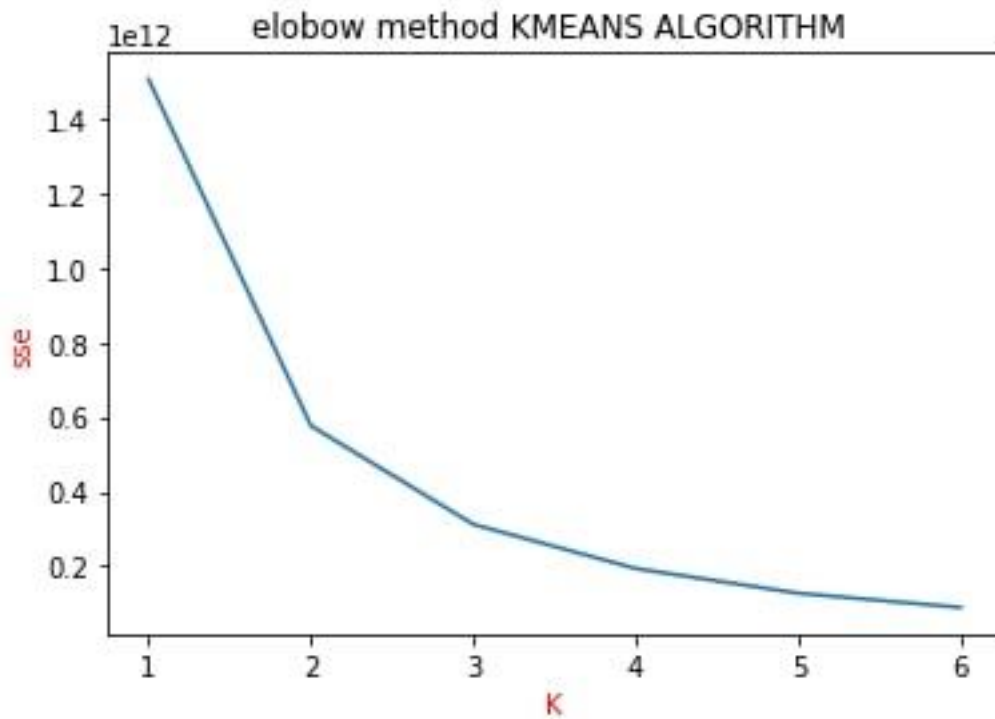
با $k=6$ تا ۲۱ تکرار به جواب رسیده است و به طور میانگین ۳۵ درصد داده‌های هر کلاستر لیبل کلاستر را دارد.

مقادیر **purity** به ازای تعداد خوشه‌های ۱ تا ۶ به صورت زیر است:

[0.17651006711409395, 0.25100671140939596, 0.2906040268456376, 0.31543624161073824, 0.34697986577181206, 0.3530201342281879]

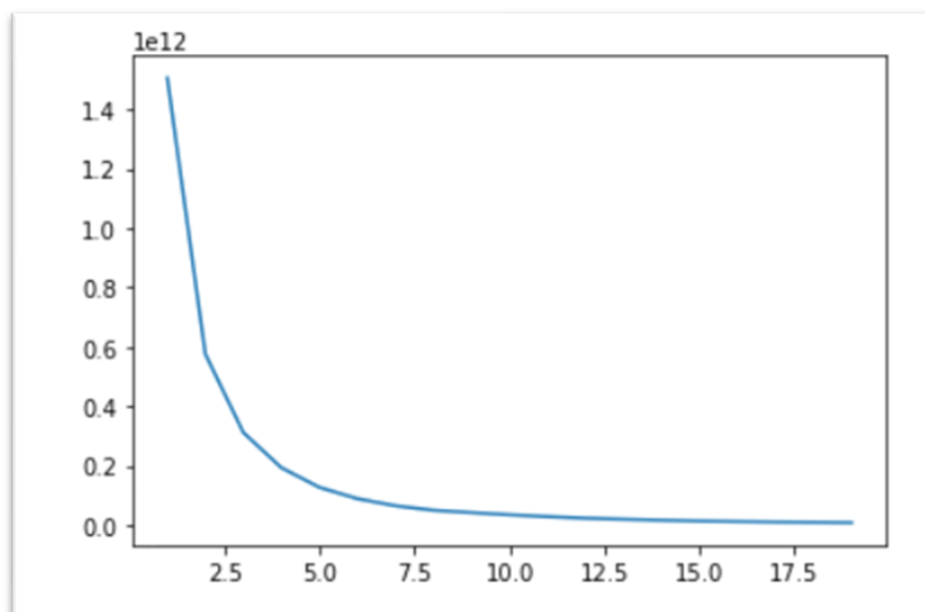
همان طور که ملاحظه می‌شود، هر چه تعداد کلاسترها بیشتر می‌شود مقدار **purity** بیشتر می‌شود و کلاسترینگ بهتری انجام می‌شود.

Elbow Method: این روش یکی از رایج ترین روش ها برای تعیین مقدار k بهینه است. به این صورت عمل می‌شود که مقدار sse برای هر لیبل محاسبه شده و با هم جمع می‌شود و برای k از ۱ تا ۶ محاسبه و پلات می‌شود.



شکل ۱۲- نمودار میزان SSE بر حسب مقدار k برای الگوریتم k -means

همان طور که مشاهده می شود میزان خطای SSE به ازای $k = 6$ کاهش چشمگیری پیدا می کند. اگر تعداد کلاستر ها را افزایش دهیم و الگوریتم را اجرا کنیم، به این صورت نمودار elbow نمایش داده می شود



شکل ۱۳- نمودار میزان SSE بر حسب مقدار k برای الگوریتم k -means

در این نمودار همان طور که مشاهده می شود بهترین K همان ۶ است و پس از آن میزان خطای sse به طور قابل توجهی کاهش نمی یابد.

الگوریتم خوشه بندی k-medoid

KMedoids مربوط به الگوریتم KMeans است. در حالی که KMeans سعی می کند مجموع مربع های درون خوشه را به حداقل برساند، KMedoids سعی می کند مجموع فواصل بین هر نقطه و Medoid خوشه خود را به حداقل برساند. Medoid یک نقطه داده است که کمترین فاصله کلی را با سایر اعضای خوشه خود دارد.

پارامتر init به صورت پیش فرض برابر Build است.

با پیاده سازی روش K-medoids به این نتایج رسیدیم:

K=۱

purity 1 : 0.17651006711409395

با $k=1$ همه ی داده ها در یک کلاستر قرار داده است. مرکز کلاستر ها در ۲۱ بعد به این صورت است که دیده می شود.

با $k=1$ مقدار purity برابر ۰.۱۷ شده است که به این معناست که ۱۷ درصد داده ها دارای لیبل هستند که بیشترین تعداد را نسبت به بقیه دارد.

با $K=1$ طبق csv که به داکيومنت پیوست شده است. نی - پیانو - سنتور - سه تار - تار - ویولن به ترتیب در ماتریس زیر آمده است.

['ney', 'piano', 'santour', 'setar', 'tar', 'violin']

[[256][240][241][263][232][258]]

با توجه به این ماتریس بیشترین عدد که ۲۶۳ است مربوط به ۴ امین است که سه تار است. پس کلاستر label سه تار گرفته است. در این قسمت چون سه تار در کل دارای تعداد داده بیشتری است برای همین تعداد آن در کلاستر بیشتر است.

K=۲

purity 2 : 0.27114093959731544

با $K=2$ ، نی - پیانو - سنتور - سه تار - تار - ویولن به ترتیب در ماتریس زیر آمده است.

['ney', 'piano', 'santour', 'setar', 'tar', 'violin']

[[127 129][235 5][72 169][168 95][187 45][133 125]]

بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی اول ۲۳۵ تاست که به ساز پیانو تعلق دارد. از این رو برچسب خوشه‌ی اول پیانو می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی دوم ۱۶۹ تا بوده که برای ساز سنتور است. از این رو برچسب خوشه‌ی دوم سنتور می‌شود.

در این قسمت سنتور label ۲ است و کلاستر ۲ هم تعداد بیشتری سنتور دارد و این نشان می‌دهد که از داده‌های کلاستر ۱ به حد خوبی فاصله داشته است. پیانو بیشتر داده‌های آن در لیبل ۱ قرار دارد و کلاستر ۱ هم تعداد بیشتری پیانو دارد و به این معناست که داده‌های آن به کلاستر ۲ شباهت خیلی کمی دارد که داده‌های سنتور در آن قرار دارد. این مورد به صورت خوبی برای تار هم برقرار است.

با $k=2$ به طور میانگین ۲۷ درصد داده‌های هر کلاستر لیبل کلاستر را دارد.

$K=3$:

purity 3 : 0.312751677852349

با $K=3$ ، نی - پیانو - سنتور - سه تار - تار - ویولن به ترتیب در ماتریس زیر آمده است.

['ney', 'piano', 'santour', 'setar', 'tar', 'violin']

[[79 85 92][226 4 10][26 122 93][89 66 108][102 12 118][68 79 111]]

بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی اول ۲۲۶ تاست که به ساز پیانو تعلق دارد. از این رو برچسب خوشه‌ی اول پیانو می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی دوم ۱۲۲ تا بوده که برای ساز سنتور است. از این رو برچسب خوشه‌ی دوم سنتور می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی سوم ۱۱۸ تا بوده که برای ساز تار است. از این رو برچسب خوشه‌ی سوم تار می‌شود.

بیشتر داده‌های سنتور در کلاستر ۲ است و نسبت به بقیه داده‌ها در کلاستر ۲ بیشتر است، در این صورت می‌توان گفت که سنتور نسبت به کلاستر ۳ و ۱ خیلی دور است. بیشتر پیانو در کلاستر ۱ است و کلاستر ۱ هم دارای تعداد بیشتری پیانو نسبت به بقیه است. با توجه به ماتریس بالا می‌توان گفت که کلاستر ۱ از کلاستر ۲ و ۳ به نسبت خوبی فاصله دارد و در بیشتر داده‌ها به خوبی جدا شده است. بیشتر داده‌های تار در کلاستر ۳ است و نسبت به بقیه داده‌ها در کلاستر ۳ بیشتر است، با توجه به ماتریس بالا می‌توان گفت که تار نسبت به کلاستر ۲ فاصله خوبی دارد و سه تار شبیه تار عمل کرده است.

با $k=3$ به طور میانگین ۳۱ درصد داده های هر کلاستر لیبل کلاستر را دارد.

$K=4$:

purity 4 : 0.3530201342281879

با $K=4$ ، نی - پیانو - سنتور - سه تار - تار - ویولن به ترتیب در ماتریس زیر آمده است.

['ney', 'piano', 'santour', 'setar', 'tar', 'violin']

[[75 53 84 44][30 2 4 204][46 82 106 7][119 41 62 41][134 4 56 38]

[90 39 96 33]]

بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی اول ۱۳۴ تاست که به ساز تار تعلق دارد. از این رو برچسب خوشه‌ی اول تار می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی دوم ۸۲ تا بوده که برای ساز سنتور است. از این رو برچسب خوشه‌ی دوم سنتور می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی سوم ۱۰۶ تا بوده که برای ساز سنتور است. از این رو برچسب خوشه‌ی سوم نیز سنتور می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی چهارم ۲۰۴ تا بوده که برای ساز پیانو است. از این رو برچسب خوشه‌ی چهارم پیانو می‌شود.

پیانو به طور خوبی از کلاستر ۱-۲-۳ فاصله دارد و بیشتر پیانو در کلاستر ۴ قرار دارد و کلاستر ۴ نسبت به بقیه داده ها دارای پیانو بیشتری است. سنتور در کلاستر ۲-۳ بیشتر است و کلاستر ۲-۳ دارای سنتور بیشتری نسبت به بقیه است که می توان گفت به صورت خوبی از کلاستر ۱-۴ جدا شده است ولی خود داده های سنتور به خوبی جدا نشده است. کلاستر ۱ که دارای label تار است با توجه به مقادیر ماتریس بالا در رابطه با تار می توان گفت از کلاستر ۲-۴ به اندازه خوبی فاصله دارد و نزدیک کلاستر ۳ است.

با $k=4$ به طور میانگین ۳۵ درصد داده های هر کلاستر لیبل کلاستر را دارد.

$K=5$:

purity 5 : 0.3516778523489933

با $K=5$ ، نی - پیانو - سنتور - سه تار - تار - ویولن به ترتیب در ماتریس زیر آمده است.

['ney', 'piano', 'santour', 'setar', 'tar', 'violin']

[[62 35 64 32 63][43 0 4 187 6][29 68 70 4 70][92 32 43 27 69]

[120 1 18 18 75][66 28 62 23 79]]

بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی اول ۱۲۰ تاست که به ساز تار تعلق دارد. از این رو برچسب خوشه‌ی اول تار می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی دوم ۶۸ تا بوده که برای ساز سنتور است. از این رو برچسب خوشه‌ی دوم سنتور می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی سوم ۷۰ تا بوده که برای ساز سنتور است. از این رو برچسب خوشه‌ی سوم نیز سنتور می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی چهارم ۱۸۷ تا بوده که برای ساز پیانو است. از این رو برچسب خوشه‌ی چهارم پیانو می‌شود. و بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی پنجم ۷۹ تا بوده که برای ساز ویولن است. از این رو برچسب خوشه‌ی چهارم ویولن می‌شود.

در این قسمت به صورت بهتری کلاستر بندی شده است. به صورتی که کلاستر ۱ دارای تعداد بیشتری تار است و تعداد بیشتر تار در کلاستر ۱ قرار گرفته است و از کلاستر های دیگر به حد خوبی فاصله دارد. کلاستر ۲-۳ سنتور است که ماتریس بالا نشان می‌دهد به صورت خوبی جدا نشده اند. کلاستر ۲-۳-۵ در داده های سنتور به شدت به هم نزدیک هستند ولی از کلاستر ۱-۴ با فاصله خوبی دور است. کلاستر ۴ دارای تعداد بیشتر پیانو است ولی با توجه به ماتریس بالا توانسته به خوبی از بقیه کلاستر ها جدا کند و فاصله خوبی با بقیه کلاستر ها داشته باشد. کلاستر ۵ دارای تعداد بیشتر نی است که به خوبی تشخیص داده شده است چون داده های نی بیشتر در کلاستر ۵ است.

با $k=5$ به طور میانگین ۳۵.۱۶ درصد داده های هر کلاستر لیبل کلاستر را دارد.

$K=6$:

purity 6 : 0.35436241610738256

با $K=6$ ، نی - پیانو - سنتور - سه تار - تار - ویولن به ترتیب در ماتریس زیر آمده است.

['ney', 'piano', 'santour', 'setar', 'tar', 'violin']

[[51 32 53 23 51 46][54 0 4 171 2 9][20 58 64 3 64 32]

[68 28 38 14 37 78][93 0 12 5 47 75][48 19 60 15 55 61]]

بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی اول ۹۳ تاست که به ساز تار تعلق دارد. از این رو برچسب خوشه‌ی اول تار می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی دوم ۵۸ تا بوده که برای ساز سنتور است. از این رو برچسب خوشه‌ی دوم سنتور می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی سوم ۶۴ تا بوده که برای ساز سنتور است. از این رو برچسب خوشه‌ی سوم نیز سنتور می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی چهارم ۱۷۱ تا بوده که برای ساز پیانو است. از این رو برچسب خوشه‌ی چهارم پیانو می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی پنجم ۶۴ تا بوده که برای ساز سنتور است. از این رو برچسب خوشه‌ی پنجم نیز سنتور می‌شود. و بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی ششم ۷۸ تا بوده که برای ساز سه‌تار است. از این رو برچسب خوشه‌ی ششم سه‌تار می‌شود.

نتایج $k=5$ و $k=6$ هردو به نسبت بهتر از مقادیر دیگر برای k است. بیشتر داده های سنتور در کلاستر ۳-۵ است ولی کلاستر ۲-۳-۵ دارای تعداد بیشتر سنتور هستند این به معنی جدا پذیری غیر دقیق است.

بیشتر داده های تار در کلاستر ۱ قرار دارد و کلاستر ۱ هم دارای بیشتر تار است و نسبت به ماتریس بالا از کلاستر های ۲-۳-۴-۵-۶ به شدت دور است. بیشتر داده های پیانو در کلاستر ۴ قرار دارد و کلاستر ۴ هم دارای بیشتر پیانو است و نسبت به ماتریس بالا از کلاستر های ۱-۲-۳-۵-۶ به شدت دور است و کلاسترینگ خوبی برای خوشه بندی آن انجام شده است. داده های سه تار بیشتر در کلاستر ۶ قرار دارد و کلاستر ۶ دارای تعداد بیشتر داده سه تار است که دقت خوبی ارائه داده است.

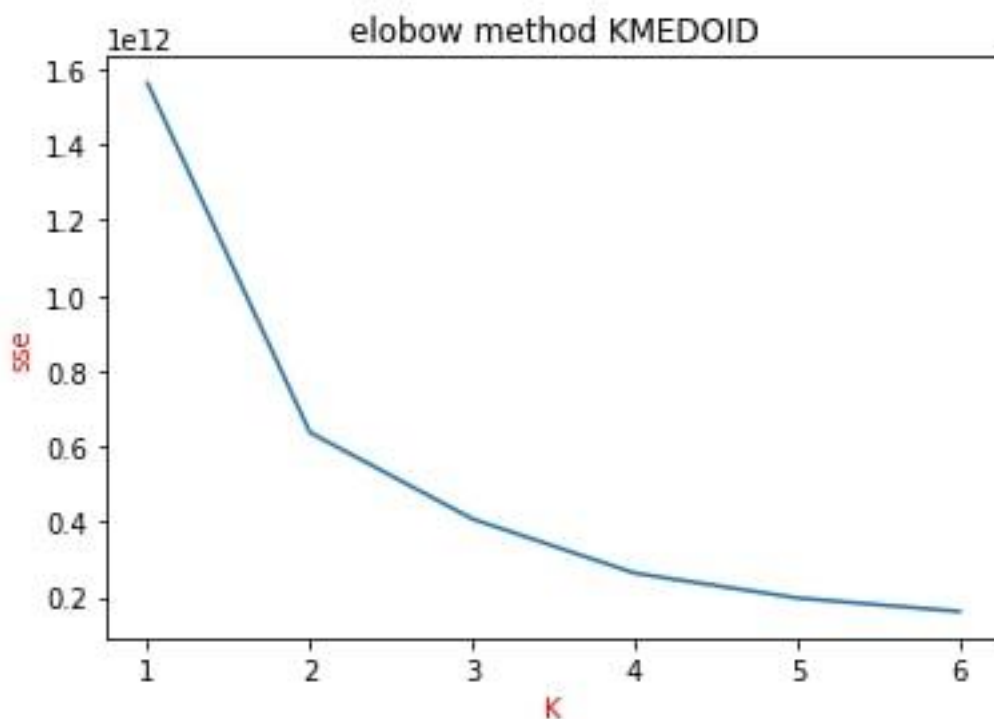
با $k=6$ به طور میانگین ۳۵.۴۳ درصد داده های هر کلاستر لیبل کلاستر را دارد.

مقادیر **purity** به ازای تعداد خوشه های ۱ تا ۶ به صورت زیر است:

[0.17651006711409395, 0.27114093959731544, 0.312751677852349, 0.3530201342281879, 0.3516778523489933, 0.35436241610738256]

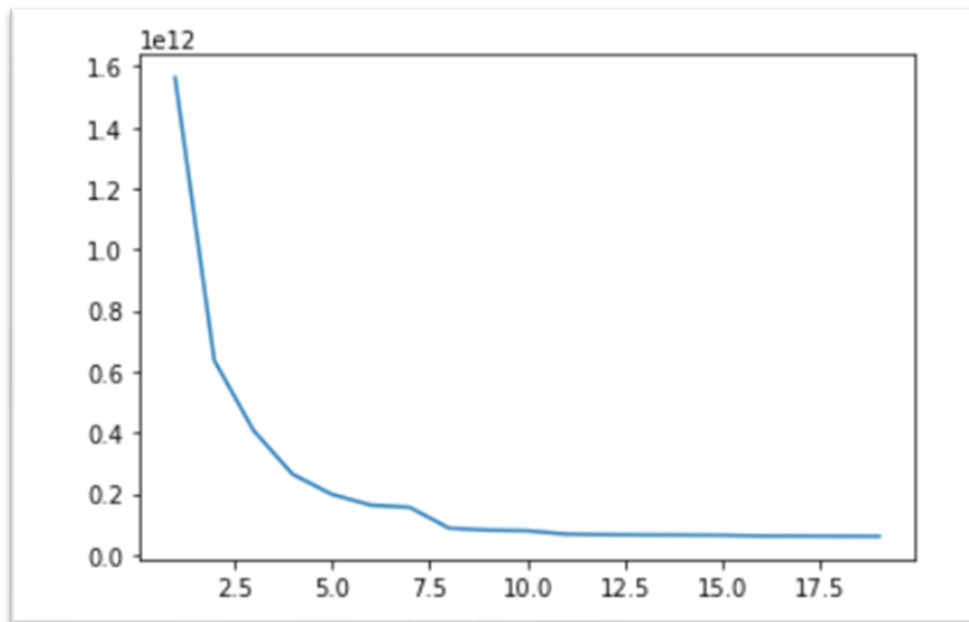
همان طور که ملاحظه می شود، هر چه تعداد کلاستر ها بیشتر می شود مقدار **purity** بیشتر می شود و کلاسترینگ بهتری انجام می شود.

Elbow Method: این روش یکی از رایج ترین روش ها برای تعیین مقدار k بهینه است. به این صورت عمل می شود که مقدار **sse** برای هر لیبل محاسبه شده و با هم جمع می شود و برای k از ۱ تا ۶ محاسبه و پلات می شود.



شکل ۱۴- نمودار میزان **SSE** برحسب مقدار k برای الگوریتم **k-medoid**

همان طور که مشاهده می شود میزان خطای SSE به ازای $k = 6$ کاهش چشمگیری پیدا می کند. اگر تعداد کلاستر ها را افزایش دهیم و الگوریتم را اجرا کنیم، به این صورت نمودار elbow نمایش داده می شود



شکل ۱۵- نمودار میزان SSE برحسب مقدار k برای الگوریتم k -medoid

در این نمودار همان طور که مشاهده می شود بهترین K همان ۶ است و پس از آن میزان خطای sse به طور قابل توجهی کاهش نمی یابد.

الگوریتم خوشه بندی Hierarchical

روش Hierarchical clustering از طریق گروه بندی داده ها در درختی از خوشه ها کار می کند. Hierarchical clustering با در نظر گرفتن هر نقطه داده به عنوان یک خوشه جداگانه آغاز می شود. سپس، به طور مکرر مراحل بعدی را اجرا می کند:

دو خوشه را که می توانند به هم نزدیکتر باشند، شناسایی می کنیم.

حداکثر ۲ خوشه قابل مقایسه را ادغام می کنیم. باید این مراحل را تا زمانی که همه خوشه ها با هم ادغام شوند ادامه دهیم.

مقدار linkage برابر average در نظر گرفته شده است. با پیاده سازی روش Hierarchical به این نتایج رسیدیم:

:K=۱

purity 1 : 0.17651006711409395

با $k=1$ همه ی داده ها در یک کلاستر قرار می گیرند. با $k=1$ مقدار purity برابر ۰.۱۷ شده است که به این معناست که ۱۷ درصد داده ها دارای لیبل هستند که بیشترین تعداد را نسبت به بقیه دارد.

با $K=1$ ، نی - پیانو - سنتور - سه تار - تار - ویولن به ترتیب در ماتریس زیر آمده است.

['ney', 'piano', 'santour', 'setar', 'tar', 'violin']

[[256],[240],[241],[263],[232],[258]]

با توجه به این ماتریس بیشترین تعداد نمونه در این تک خوشه مربوط به ساز سه تار با مقدار ۲۶۳ می باشد. البته از آنجایی که در مجموعه داده پیش پردازش شده تعداد نمونه های ساز سه تار بیشتر است این اتفاق رخ داده است.

:K=۲

purity 2 : 0.2295302013422819

با $K=2$ ، نی - پیانو - سنتور - سه تار - تار - ویولن به ترتیب در ماتریس زیر آمده است.

['ney', 'piano', 'santour', 'setar', 'tar', 'violin']

[[3 253],[83 157],[2 239],[4 259],[1 231],[4 254]]

بیشترین تعداد داده ی مربوط به یک ساز در خوشه ی اول ۸۳ تا است که به ساز پیانو تعلق دارد. از این رو برچسب خوشه ی اول پیانو می شود. و بیشترین تعداد داده ی مربوط به یک ساز در خوشه ی دوم ۲۵۹ تا بوده که برای ساز سه تار است. از این رو برچسب خوشه ی دوم سه تار می شود.

در این قسمت سه تار label ۲ است و کلاستر ۲ هم تعداد بیشتری سه تار دارد و این نشان می دهد که از داده های کلاستر ۱ به حد خوبی فاصله داشته است. پیانو بیشتر داده های آن در لیبل ۲ قرار دارد ولی کلاستر ۱ تعداد بیشتری پیانو نسبت به بقیه داده ها دارد.

در تمامی داده ها مقادیر آن به کلاستر ۲ بیشتر از کلاستر ۱ نزدیک است.

با $k=2$ به طور میانگین ۲۲ درصد داده های هر کلاستر لیبل کلاستر را دارد.

K=۳:

purity 3 : 0.2295302013422819

با $K=۳$ ، نی - پیانو - سنتور - سه تار - تار - ویولن به ترتیب در ماتریس زیر آمده است.

['ney', 'piano', 'santour', 'setar', 'tar', 'violin']

[[253 3 0][157 81 2][239 2 0][259 4 0][231 1 0][254 4 0]]

بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی اول ۲۵۹ تاست که به ساز سه‌تار تعلق دارد. از این رو برچسب خوشه‌ی اول سه‌تار می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی دوم ۸۱ تا بوده که برای ساز پیانو است. از این رو برچسب خوشه‌ی دوم پیانو می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی سوم ۲ تا بوده که برای ساز پیانو است. از این رو برچسب خوشه‌ی سوم نیز پیانو می‌شود.

بیشتر داده‌ها غیر از پیانو در کلاستر ۱ است و نسبت به بقیه کلاسترها خیلی دور هستند. بیشتر پیانو در کلاستر ۱ است ولی کلاستر ۲ دارای تعداد بیشتری پیانو نسبت به بقیه است. با توجه به ماتریس بالا می‌توان گفت که کلاستر ۱ از کلاستر ۲-۳ به نسبت خوبی فاصله دارد ولی در بیشتر داده‌ها کلاستر ۱ به خوبی جدا نشده‌اند.

با $K=۳$ به طور میانگین ۲۲ درصد داده‌های هر کلاستر لیبل کلاستر را دارد.

K=۴:

purity 4 : 0.30604026845637583

با $K=۴$ ، نی - پیانو - سنتور - سه تار - تار - ویولن به ترتیب در ماتریس زیر آمده است.

['ney', 'piano', 'santour', 'setar', 'tar', 'violin']

[[3 32 0 221][81 145 2 12][2 11 0 228][4 59 0 200][1 51 0 180]

[4 47 0 207]]

بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی اول ۸۱ تاست که به ساز پیانو تعلق دارد. از این رو برچسب خوشه‌ی اول پیانو می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی دوم ۱۴۵ تا بوده که برای ساز پیانو است. از این رو برچسب خوشه‌ی دوم نیز پیانو می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی سوم ۲ تا بوده که برای ساز پیانو است. از این رو

برچسب خوشه‌ی سوم نیز پیانو می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی چهارم ۲۲۸ تا بوده که برای ساز سنتور است. از این رو برچسب خوشه‌ی چهارم سنتور می‌شود.

پیانو به طور خوبی از کلاستر ۱-۳-۴ فاصله دارد و بیشتر پیانو در کلاستر ۲ قرار دارد و کلاستر ۲ نسبت به بقیه داده ها دارای پیانو بیشتری است. سنتور در کلاستر ۴ بیشتر است و کلاستر ۴ دارای سنتور بیشتری نسبت به بقیه است. کلاستر ۱ که دارای label پیانو است با توجه به مقادیر ماتریس بالا در رابطه با پیانو می توان گفت از کلاستر ۳-۴ به اندازه خوبی فاصله دارد. با $k=4$ به طور میانگین ۳۰ درصد داده های هر کلاستر لیبل کلاستر را دارد.

$K=5$:

purity 5 : 0.30604026845637583

با $K=5$ ، نی - پیانو - سنتور - سه تار - تار - ویولن به ترتیب در ماتریس زیر آمده است.

['ney', 'piano', 'santour', 'setar', 'tar', 'violin']
 [[1 2 0 221 32][38 43 2 12 145][1 1 0 228 11][3 1 0 200 59]
 [0 1 0 180 51][3 1 0 207 47]]

بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی اول ۳۸ تاست که به ساز پیانو تعلق دارد. از این رو برچسب خوشه‌ی اول پیانو می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی دوم ۴۳ تا بوده که برای ساز پیانو است. از این رو برچسب خوشه‌ی دوم نیز پیانو می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی سوم ۲ تا بوده که برای ساز پیانو است. از این رو برچسب خوشه‌ی سوم هم پیانو می‌شود. بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی چهارم ۲۲۸ تا بوده که برای ساز سنتور است. از این رو برچسب خوشه‌ی چهارم سنتور می‌شود و بیشترین تعداد داده‌ی مربوط به یک ساز در خوشه‌ی پنجم ۱۴۵ تا بوده که برای ساز پیانو است. از این رو برچسب خوشه‌ی پنجم نیز پیانو می‌شود.

در این قسمت به صورت بهتری کلاستر بندی شده است. به صورتی که کلاستر ۱ دارای تعداد بیشتری پیانو است و تعداد بیشتر پیانو در کلاستر ۵ قرار گرفته است و این نشان دهنده دقت پایین است. کلاستر ۲-۴ سنتور است که ماتریس بالا نشان می دهد به صورت خوبی جدا نشده اند. کلاستر ۴ دارای تعداد بیشتر سنتور است ولی با توجه به ماتریس بالا صدای تار و سه تار و ویولن و سنتور و نی به هم خیلی شبیه در نظر گرفته شده است و نتوانسته به خوبی آن ها را از هم جدا کند. کلاستر ۵ دارای تعداد بیشتر پیانو است که به خوبی داده های آن از بقیه کلاستر ها جدا شده است.

با $k=5$ به طور میانگین ۳۰ درصد داده های هر کلاستر لیبل کلاستر را دارد.

K=6:

purity 6 : 0.30604026845637583

با $K=6$ ، نی - پیانو - سنتور - سه تار - تار - ویولن به ترتیب در ماتریس زیر آمده است.

['ney', 'piano', 'santour', 'setar', 'tar', 'violin']

[[2 32 0 221 1 0][43 145 5 12 33 2][1 11 0 228 1 0][1 59 0 200 3 0]

[1 51 0 180 0 0][1 47 0 207 3 0]]

باتوجه به مقادیر نشان داده شده در ماتریس‌های بالا خوشه‌های اول و دوم و سوم و پنجم و ششم برچسب پیانو می‌گیرند. در حالیکه خوشه‌ی چهارم برچسب سنتور می‌گیرد.

نتایج $k=5$ و $k=6$ هر دو به نسبت بهتر از مقادیر دیگر برای k است. بیشتر داده‌های نی و سنتور و سه تار و تار و ویولن در کلاستر ۴ است و کلاستر ۱-۲-۳-۵-۶ هم دارای بیشتر پیانو است و باتوجه به ماتریس بالا به ۲ دسته تقسیم شده است کلاستر ۴ که همه داده‌ها غیر پیانو است و کلاستر دیگر که پیانو است.

به این صورت با دقت خیلی کمی کلاسترینگ شده است.

با $k=6$ به طور میانگین ۳۰ درصد داده‌های هر کلاستر لیبل کلاستر را دارد.

مقادیر purity به ازای تعداد خوشه‌های ۱ تا ۶ به صورت زیر است:

[0.17651006711409395, 0.2295302013422819, 0.2295302013422819, 0.30604026845637583, 0.30604026845637583, 0.30604026845637583]

همان طور که ملاحظه می‌شود، هر چه تعداد کلاستر ها بیشتر می‌شود مقدار purity بیشتر می‌شود و کلاسترینگ بهتری انجام می‌شود.

سایر روش‌ها:

DBSCAN Algorithm: با توجه به سختی تنظیم پارامتر ها این روش دارای جواب معتبر نشد.

Meanshift Algorithm: این الگوریتم در درس مطرح نشده بود ولی پیاده سازی کردیم.

نتیجه گیری:

با توجه به تحلیل هایی که کردیم به این نتیجه رسیدیم که ۲ روش k-means و k-medoids به هم خیلی شبیه هستند ولی روش k-medoids عملکرد بهتری داشت. روش Hierarchical خیلی غیر دقیق تر عمل کرد.

با توجه به جدول ۱ می توان مشاهده کرد که خوشه بندی به روش k-medoid بهترین عملکرد (بر اساس معیار purity) را نسبت به روش های k-means و hierarchical دارد. دلیل این موضوع می تواند این باشد که روش k-medoid نسبت به outlier حساس نمی باشد. همچنین می توان مشاهده کرد که با افزایش تعداد خوشه ها عملکرد خوشه بندیها (از نظر معیار purity) بهتر می شود. همانطور که انتظار میرفت روش های clustering بر روی داده های ما عملکرد خیلی خوبی نداشتن زیرا از طرفی داده ها درون هر دسته واریانس زیاد و بین دسته ها واریانس کم بوده و همچنین نویز های زیادی در داخل آهنگ های مختلف وجود داشت و از طرفی نامتوازن بودن داده ها باعث ضعف عملکرد clustering ها میشد.

جدول ۱ - مقدار purity به ازای kهای مختلف در روش های مختلف خوشه بندی

	K=1	K=2	K=3	K=4	K=5	K=6
k-means	0.176	0.251	0.290	0.315	0.346	0.353
k-medoid	0.176	0.271	0.312	0.353	0.351	0.354
hierarchical	0.176	0.229	0.229	0.306	0.306	0.306

جدول ۲ - برچسب خوشه ها به ازای kهای مختلف در روش های مختلف خوشه بندی

	K=2		K=3			K=4			
k-means	سنتور	پیانو	سنتور	سنتور	پیانو	سنتور	سنتور	سنتور	پیانو
k-medoid	پیانو	سنتور	پیانو	سنتور	تار	تار	سنتور	سنتور	پیانو
hierarchical	پیانو	سه تار	سه تار	پیانو	پیانو	پیانو	پیانو	پیانو	سنتور

جدول ۳ - ادامه ی جدول ۲

	K=5				
k-means	پیانو	سنتور	سنتور	تار	نی
k-medoid	تار	سنتور	سنتور	پیانو	نی
hierarchical	پیانو	سنتور	نی/پیانو	سنتور	پیانو

	K = 6					
k-means	ویولن	سنتور	تار	سنتور	پیانو	نی
k-medoid	تار	سنتور	سنتور	پیانو	سنتور	سه‌تار
hierarchical	پیانو	پیانو	پیانو	سنتور	پیانو	پیانو