

به نام خدا



دانشگاه تهران

دانشکده فنی

دانشکده مهندسی برق و کامپیوتر



گزارش نهایی پروژه یادگیری ماشین

اعضای گروه :

محمد صادق صادقی

حامد غلامی

امیرحسین محمدی

حسین خلیلی

دی ماه ۱۴۰۰

فهرست مطالب

۲	۱	مقدمه
۳	۲	استخراج ویژگی‌ها و پیش‌پردازش داده‌ها
۳	۱.۲	ویژگی‌های حوزه زمان
۳	۲.۲	ویژگی‌های حوزه کپسترال
۴	۳.۲	ویژگی‌های حوزه فرکانس
۶	۴.۲	ضرایب کپسترال مبتنی بر فرکانس مل
۸	۵.۲	انتخاب ویژگی‌ها
۹	۳	طبقه‌بندی
۹	۱.۳	طبقه‌بندی به کمک شبکه MLP
۱۳	۲.۳	طبقه‌بندی به کمک شبکه XGBoost
۱۵	۳.۳	طبقه‌بندی به کمک شبکه SVM
۱۸	۴	خوشه‌بندی
۱۸	۱.۴	استفاده از روش PCA برای کاهش ابعاد
۱۹	۲.۴	خوشه‌بندی به روش k نزدیکترین مرکز
۲۲	۳.۴	خوشه‌بندی به روش سلسله مراتبی
۲۴	۵	ضمیمه: نحوه اجرای برنامه و مشاهده خروجی

۱ مقدمه

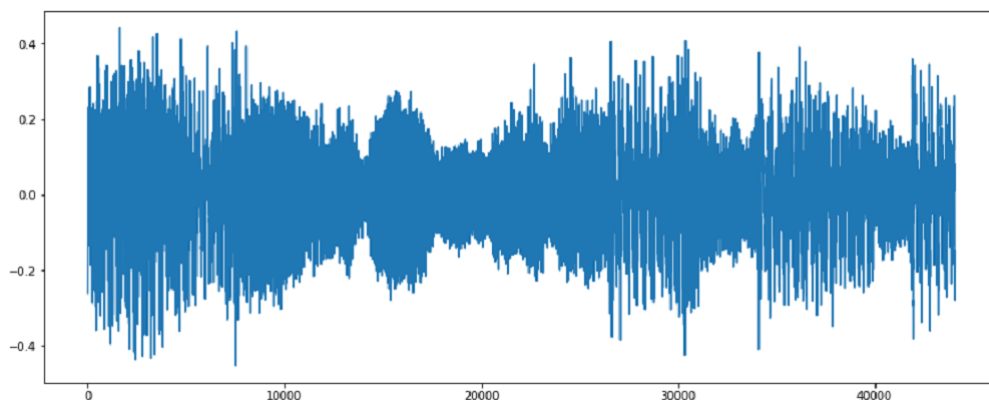
در این پروژه هدف ما بررسی الگوریتم‌های طبقه‌بندی و خوشه‌بندی برای ۵ دسته از اهنگ‌های لری، کردی، گیلکی، بندری و ترکی می‌باشد. به این منظور ما در گام اول، نحوه استخراج ویژگی‌ها و پیش‌پردازش داده‌های استخراج شده را به طور کامل توضیح می‌دهیم و در مورد هر دسته از ویژگی‌ها به طور خلاصه اطلاعاتی را بیان می‌کنیم. در بخش بعدی، سه الگوریتم طبقه‌بندی پرسپترون چندلایه، XGBoost و SVM را بر روی داده‌های بدست آمده از گام قبلی اجرا کرده و نتایج بدست آمده را مورد تحلیل و بررسی قرار می‌دهیم. در این گام نتایج بدست آمده نشان دهنده عملکرد مناسب مدل‌های پیاده‌سازی شده می‌باشد، در گام آخر نیز دو الگوریتم خوشه‌بندی Kmeans و Agglomerative به منظور خوشه‌بندی مجموعه داده‌ها استفاده می‌شود و نتایج بدست آمده مورد تحلیل و بررسی قرار می‌گیرد.

۲ استخراج ویژگی ها و پیش پردازش داده ها

برای ساختن طبقه بند ، نیاز به ویژگی های متفاوت از آهنگ های داده شده داریم. حال برای اینکه بتوانیم این ویژگی ها را استخراج کنیم یعضا نیاز است که به حوزه فرکانس موسیقی ها علاوه بر حوزه زمان موسیقی ها توجه کنیم (و در مواردی هر دوی آن ها). به طور خلاصه ویژگی ها را میتوان به دسته های زیر تقسیم کرد:

۱.۲ ویژگی های حوزه زمان

نرخ عبور از صفر (Zero Crossing Rate) ، میزان جذر میانگین مربعات انرژی RMSE، انرژی کوتاه مدت (STE) ، و آنتروپی شانون



شکل ۱: Audio Signal in Time Domain (in one second) as we can see there can not be much be understood from raw time domain.

۲.۲ ویژگی های حوزه کپسترال

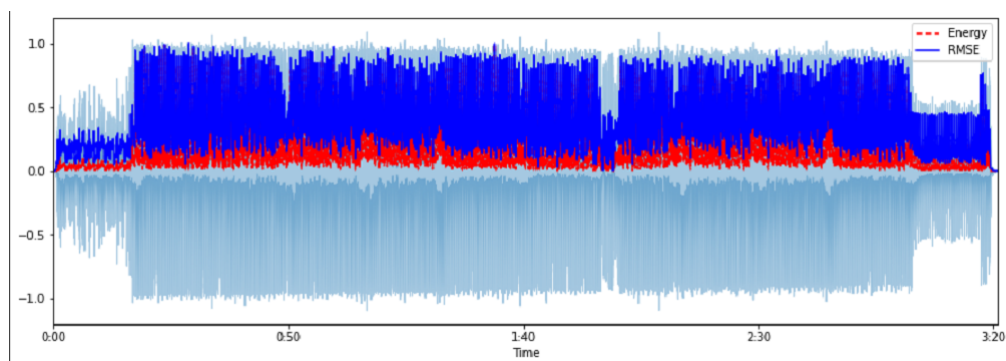
ضرایب کپسترال مبتنی بر فرکانس مل (Mel Frequency Cepstral Coefiecients MFCC) ، لگاریتم مل اسپکتروگرام (log melspec) توجه شود که موسیقی ها فایل هایی با طول زیاد است در نتیجه در خیلی از موازد ویژگی های بالا، داده ها ابتدا به فریم های ۴۰ میلی ثانیه ای با ۱۰ میلی ثانیه همپوشانی با فریم بعدی (برای جلوگیری از اثرات پنجره کردن سیگنال) تبدیل شده و سپس روش ها اعمال شده تا به خروجی معنا دار تری برسیم.

حال به توضیح مختصر از ویژگی های به دست آمده می پردازیم. ویژگی های حوزه زمان:

الف) نرخ عبور از صفر: نرخ تغییر سیگنال از مثبت به منفی و برعکس. حاوی اطلاعات خوبی برای آهنگ های ضربی میباشد میتوان به وسیله زیر آن را فرموله کرد.

$$z_{CR} = 1/(T - 1) \sum_{t=1}^{T-1} l_{R<0}(s_t s_{t-1})$$

ب) جذر میانگین مربعات انرژی: نشان دهنده توان سیگنال است. مقدار کوتاه مدت آن انرژی هر پنجره میباشد.



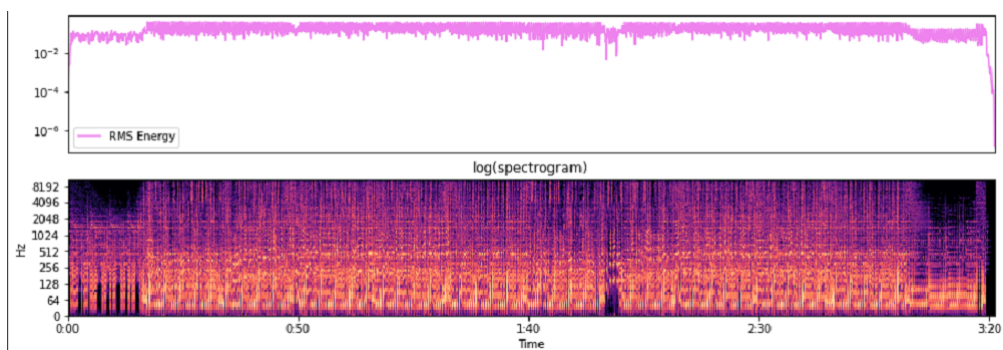
شکل ۲: Root Mean Square Energy, Signal in Time Domain, and Energy for one song

ج) آنتروپی شانون: معیاری برای نشان دادن میزان اطلاعاتی که در سیگنال وجود دارد.

۳.۲ ویژگی های حوزه فرکانس

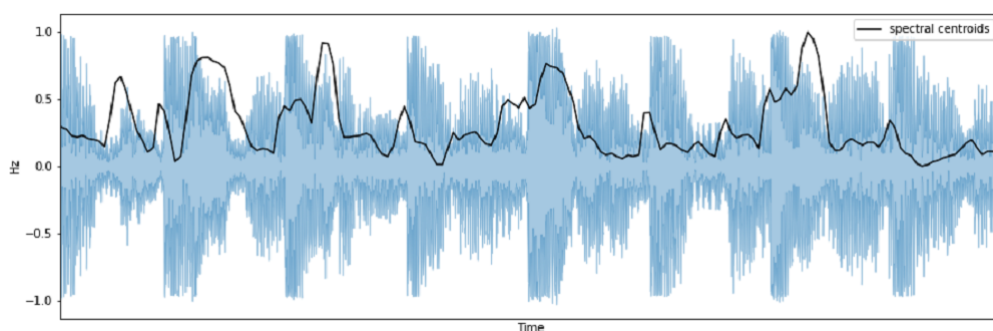
اندازه طیف سیگنال، فوریه کوتاه مدت از کروما (Chroma-STFT)

اندازه طیف سیگنال: طیف سیگنال نشان دهنده میزان انرژی در حوزه فرکانس است، به عبارت دیگر میتوان از این مقدار متوجه شد که در هر فرکانس به خصوص چه میزان انرژی وجود دارد. فوریه کوتاه زمان STFT در این تبدیل همانطور که در بالا تر اشاره شد، سیگنال حوزه زمان را به فریم های کوتاه مدت تقسیم میکنیم و از هر فریم تبدیل فوریه میگیریم. با اینکار میتوان یک دید زمان-فرکانس به سیگنال داشت. برای مشاهده بهتر اسپکتروگرام یک موسیقی در شکل زیر به همراه RMS رسم شده است.



شکل ۳: Log Power Spectrogram and RMS Energy, As we can see Spectrogram gives us a intuition in both frequency and Time Domain while RMS Energy is only in Time Domain.

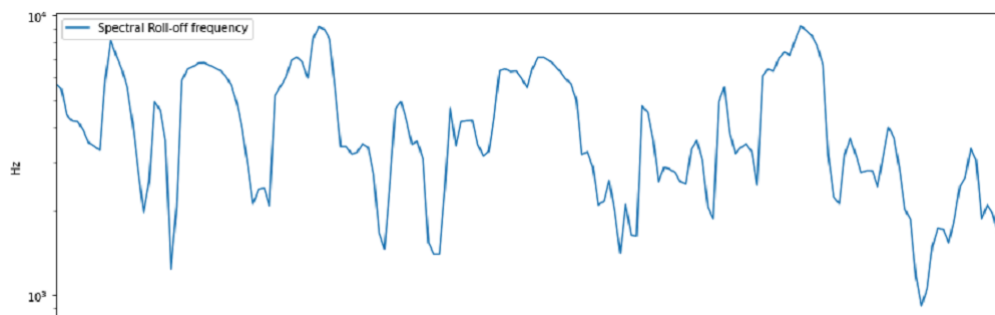
STFT میانگین (mean) و چولگی (skewness) آن را به فیچرها اضافه میکنیم چرا که میتواند نشان دهنده تمپو سیگنال ما باشد. در شکل زیر Spectral Centorid یا همان میانگین اسپکترال برای هر فریم و همچنین رسم شده است.



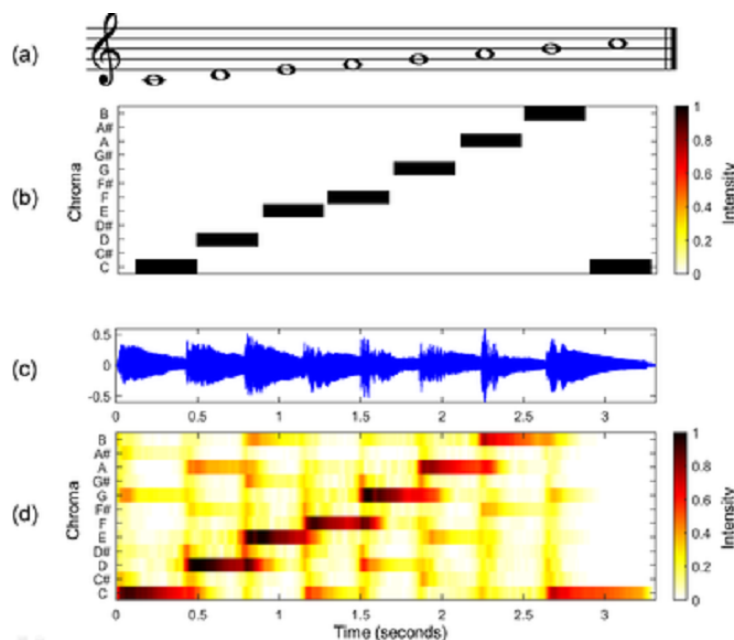
شکل ۴: Spectral Centroids and Time Domain Music.

Chroma STFT : کروما در حقیقت نگاه به موسیقی از دید ۱۲ pitch میباشد. ابزاری قوی برای تحلیل موسیقی میباشد و به طور معنا داری به ۱۲ سطح با گام مساوی تقسیم میکند. میتواند ویژگی هارمونیکی موسیقی را در خود نمایش بدهد.

در شکل ۸ توان اسپکتروگرام کروما برای یک موسیقی به مدت ۴ ثانیه به نمایش گذاشته شده است.



شکل ۵: Spectral Centroids and Time Domain Music.

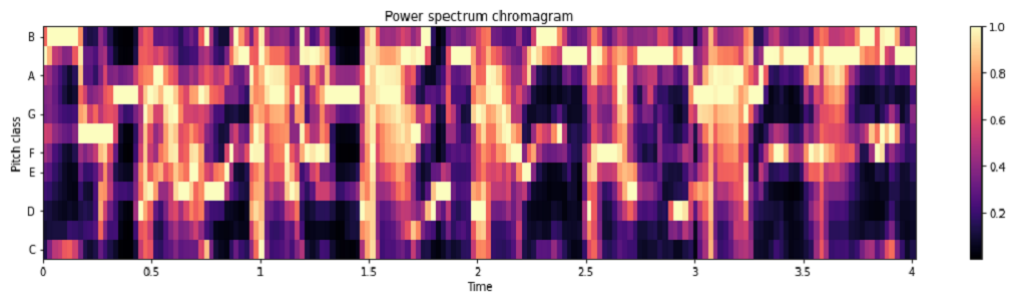


شکل ۶: Chroma Feature

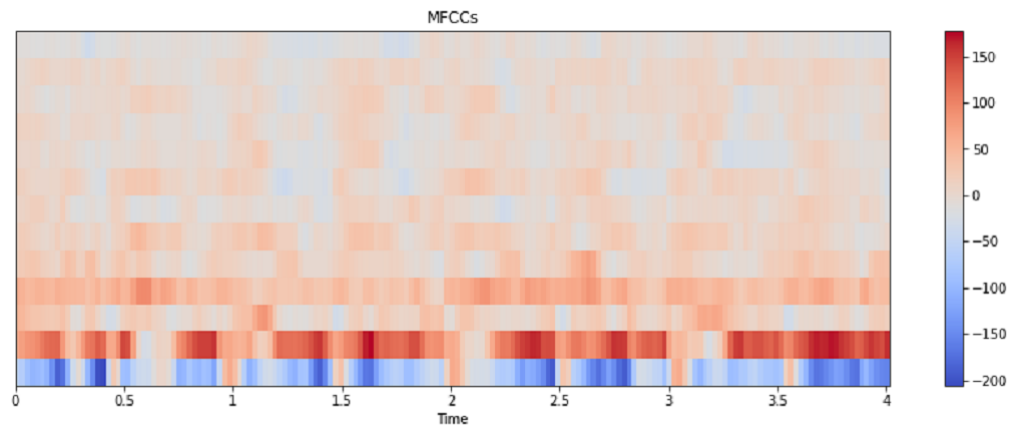
۴.۲ ضرایب کپسترال مبتنی بر فرکانس مل

روشی است برای به دست آوردن ویژگی های صوتی انسان، اگر موسیقی بی کلام باشد نمیتواند ویژگی های خوبی به دست آورد اما در غیر این صورت عملکرد بسیار خوبی دارد.

حال با محاسبه همه ویژگی های فوق به وسیله کتابخانه های librosa و surfboard میتوان از ویژگی ها استفاده کرد. اما با توجه به اینکه ما هر آهنگ را به فریم های ۵۰ میلی ثانیه تبدیل



شکل ۷: Power Spectrum Chromagram, We can See How different Pitches have different powers.



شکل ۸: MFCC for clipped audio Signal

میکنیم و حدود ۱۰۰۰ آهنگ داریم و هر آهنگ به طور متوسط ۳ دقیقه طول دارد منوجه می‌شویم که تعداد فیچر ها عددی بسیار بزرگ میشد، برای ذخیره کردن بیشترین اطلاعات ممکن را از فریم های تبدیل شده، میتوان نماینده هایی را مانند میانگین، واریانس، فاصله چارک اول و سوم و چولگی آن را انتخاب کنیم. لازم به ذکر است که این چهار نماینده برای همه ی چهار ویژگی استفاده نشده و به نسبت اهمیت استفاده شده است.

```
genres = 'Lori Kordi Torki Bandari Gilaki'.split()
for g in genres:
    # for filename in os.listdir(f'./dataset/ML_Data G1/{g}'):
    _, _files = next(os.walk(f'./ML_Data G1/{g}'))
    for filename in _files:
        songname = f'./ML_Data G1/{g}/{filename}'
        sound = Waveform(path=songname)
        main1.append(extract_features([sound],
                                     ['mfcc', 'log_melspec'],
                                     ['mean', 'std', 'skewness',
                                      'kurtosis'])))
        main2.append(extract_features([sound], ['rms', 'chroma_stft',
                                                'zerocrossing', 'ppe'],
                                     ['mean', 'std'])))
        main3.append(extract_features([sound], ['magnitude_spectrum'],
                                     ['mean', 'std', 'skewness',
                                      'kurtosis', 'q3_q1_range'])))
        main4.append(extract_features([sound], ['shannon_entropy_slidingwindow'],
                                     ['mean', 'std', 'skewness', 'kurtosis'])))
```


فیچر های فوق به وسیله کد بالا به دست می آید. حال بعد از اجرای برنامه متوجه می شویم که برخی از داده ها نرخ نمونه برداری برای سازگاری با کتاب خانه های ما را ندارند در نتیجه باید پاکسازی شوند. آهنگ های زیر حذف شده اند.

```
!rm -rf "/ML_Data G1/Lor1/89.mp3"
!rm -rf "/ML_Data G1/Gilaki/11.mp3"
!rm -rf "/ML_Data G1/Bandari/22.mp3"
!rm -rf "/ML_Data G1/Torki/104.mp3"
!rm -rf "/ML_Data G1/Lor1/156.mp3"
```

در مرحله بعد متوجه میشویم که برخی از ویژگی ها حاوی NaN میباشد. میتوان به کمک کتابخانه scikit learn این ویژگی ها را با میانه آن ها جایگزین کرد.

```
[ ]: from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy="median")
imputer.fit(X_train)

X_train[:] = imputer.transform(X_train)
X_test[:] = imputer.transform(X_test)
```

حال که ویژگی ها استخراج شده اند میتوان بهترین هایشان را برای آموزش classifier انتخاب کرد.

۵.۲ انتخاب ویژگی ها

برای انتخاب ویژگی دو روش کلی وجود دارد wrapper methods و filter methods استفاده کرد. با اعمال PCA و LDA که جزوی از Filter method ها هستند مشاهده می کنیم که نه تنها عملکرد طبقه بند بهتر نمی شود که به شدت بدتر می شود برای جست و جوی علت این پدیده می توان به روش انتخاب ویژگی در این دو الگوریتم توجه کرد. الگوریتم PCA سعی در حفظ هرچه بیشتر variation در داده ها می کند و این کار را با انتخاب بردار ویژه هایی که متناظر با بزرگترین مقدارهای ویژه هستند انجام می دهد که این را می توان به پایین بودن SNR (نرخ سیگنال به نویز) نسبت داد زیرا بیشترین variation هم شامل نویز می شود و هم شامل سیگنال که تاثیر بالایی در این variation باعث عملکرد ضعیف مدل می شود. به عبارت دیگر PCA زمانی میتواند خوب عمل کند که PC های انتخاب شده بتواند variation داده ها را به خوبی بپوشاند اما در مورد ما نمیتواند. با استفاده از Wrapper Method ها دقت طبقه بند به صورت کلی بهبود می یابد که این نکته را به طور کلی می توان به این نسبت داد که این روش ها مجموعه ی ویژگی ها را تا جایی انتخاب می کنند که عملکرد بهبود بیابد و این که این روش ها به طور کل بود یا نبود یک ویژگی را مد نظر میگیرند که این روش به نویز مقاوم تر است، هر چند عمومیت (Genrality) کمتری نسبت به filter methods دارند و همچنین از لحاظ پیچیدگی زمانی نیز بدتر است اما نتیجه بهتری را میدهد. در این راستا ۱۰۳۴ فیچر که بهترین عملکرد را داشتند انتخاب شدند. خروجی متناظر بعد از اعمال کردن روش های فوق با الگوریتم های متفاوت در قسمت بعد مشاهده می شود.

۳ طبقه‌بندی

در این بخش، ما به منظور طبقه‌بندی موسیقی‌های محلی از سه مدل مختلف پرسپترون چندلایه، XGBoost و SVM استفاده کردیم و هر کدام از مدل‌ها را براساس داده‌های پیش‌پردازش شده آموزش می‌دهیم و در نهایت به تحلیل و بررسی نتایج بدست آمده از این روش‌ها می‌پردازیم و نتایج را در قالب نمودارهایی ارائه می‌دهیم.

۱.۳ طبقه‌بندی به کمک شبکه MLP:

در این قسمت، هدف ما این است که براساس ویژگی‌های انتخاب شده یک، یک مدل پرسپترون چند لایه را آموزش بدهیم. به منظور ایجاد این مدل ما از کتابخانه‌ی Pytorch استفاده کرده‌ایم و لازم است که پیش بیان نتایج بدست آمده، توضیحات مختصری در مورد تعداد لایه‌های استفاده شده و پارامترهای به کار گرفته شده بیان گردد.

لازم به ذکر است که ما از بین اهنک‌های موجود ده درصد را برای تست و ده درصد را بیان اعتبار سنجی و هشتاد درصد را به منظور آموزش در نظر گرفتیم.

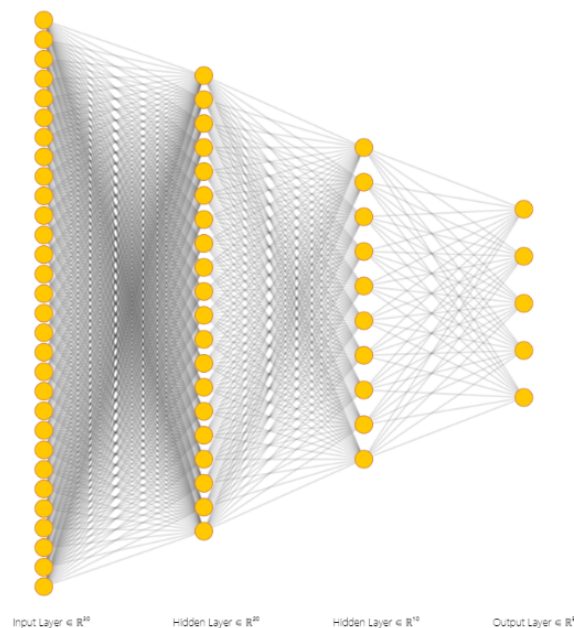
شمای کلی مدل مورد استفاده:

برای این قسمت، با توجه به حجم بالای ویژگی‌های استخراج شده، ما از کتابخانه Sklearn و تابع SelectFromModel به منظور انتخاب ویژگی‌های مناسب استفاده کردیم و تخمینگر را براساس الگوریتم SVM با پارامتر $C = 0.02$ قرار دادیم بعد از اعمال این الگوریتم از میان ویژگی‌های موجود، ۳۶۶ ویژگی پراهمیت انتخاب شد که ادامه‌ی کار را براساس این ویژگی‌ها انجام می‌دهیم.

اکنون پس از آماده سازی مجموعه داده، به توضیح مدل ساخته شده می‌پردازیم، ما در این مدل در لایه اول به تعداد ویژگی‌های انتخاب شده یعنی ۳۶۶ نورون قرار می‌دهیم، در لایه دوم و سوم به ترتیب ده و بیست نورون قرار داده و لایه آخر به این دلیل که وظیفه‌ی ما دارای ۵ برچسب است، از ۵ نورون استفاده می‌کنیم تا در نهایت نورون که بالاترین مقدار را داشت نشان دهنده‌ی نوع موسیقی باشد. لازم به ذکر است که در میان این شبکه‌ها ما از لایه‌های drop out به منظور پیشگیری از overfit استفاده کرده‌ایم. همچنین تابع فعالساز مورد استفاده در لایه دوم و سوم Relu و در لایه‌ی آخر از LogSoftmax استفاده شده است. در شکل ۹ شمای کلی شبکه مورد استفاده نشان داده شده است.

پارامترها و الگوریتم بهینه‌سازی استفاده شده:

به منظور انجام این وظیفه ما مقدار drop out را برابر ۵۰ درصد در نظر گرفته‌ایم همچنین

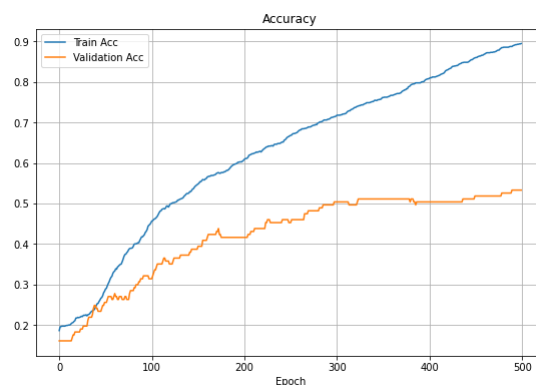


شکل ۹: MLP Structure

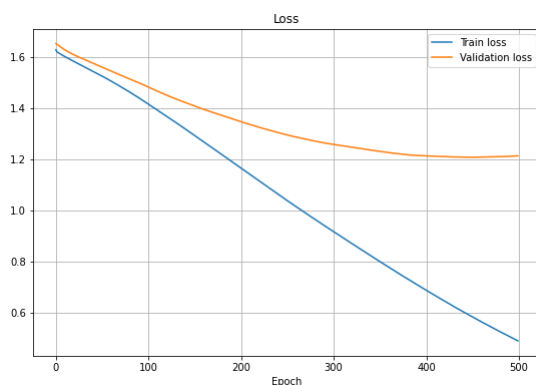
با توجه به ماهیت که تابع هزینه مورد استفاده Cross Entropy می‌باشد. تابع بهینه‌سازی مورد استفاده نیز Adam است که نرخ یادگیری آن را 0.001 در نظر گرفتیم و در مجموعه ما یادگیری را در 500 Epoch انجام می‌دهیم. پس از بیان این مقدمات اکنون لازم است که در خصوص نتایج بدست آمده توضیحات بیان گردد. نتایج بدست آمده و تحلیل‌ها:

همانطور که بیان شد ما آموزش را در 500 اپاک و تا زمانی که بیش‌برازش رخ ندهد ادامه دادیم و در نهایت نمودار مربوط به میزان loss بدست آمده برای دو داده‌ی اعتبارسنجی و آموزش نسبت به تعداد اپاک‌ها مطابق شکل زیر شد. همانطور که در این نمودار به وضوح مشخص است، با افزایش اپاک‌ها مقدار Loss در هر دو داده با کاهش همراه است و این نکته به وضوح مشخص است که الگوریتم ما به هیچ وجه دچار بیش‌برازش نشده است و در جایی که روند کاهشی به یک روند پایدار تبدیل شد و ما آموزش را متوقف کردیم.

اکنون به منظور بررسی عملکرد کار، یک نمودار براساس میزان دقت بدست آمده در هر اپاک برای دو داده‌ی اعتبارسنجی و آموزش رسم کرده‌ایم. همانطور که در این نمودار مشخص است، دقت داده‌های آموزش تا حدود 90 درصد می‌رسد اما دقت داده‌های اعتبارسنجی در این 500 اپاک بیشتر از 55 درصد نمی‌شود.



شکل ۱۰: Loss function



شکل ۱۱: Accuracy

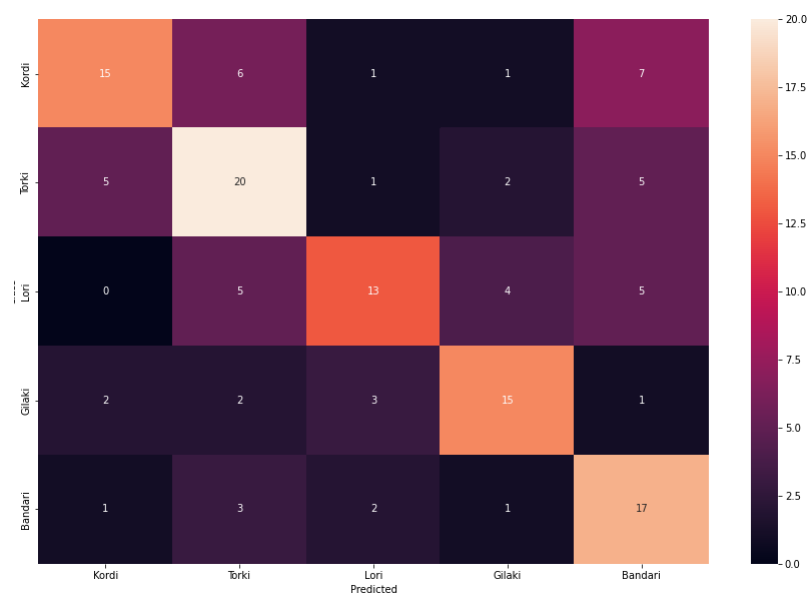
پس از پایان مرحله‌ی آموزش مدل، اکنون مدل توسط داده‌های آموزش مدل آزمایش قرار می‌گیرد. به این منظور داده‌های تست موجود را به ورودی مدل اعمال کرده و پیش‌بینی نوع آهنگ‌های بر آن اساس صورت می‌گیرد و با توجه به برجسب اصلی داده‌های معیارهای دقت و F1 score(macro) برای نتایج بدست آمده محاسبه می‌شود. که نتایج آن مطابق جدول زیر است:

مقدار بدست آمده:	معیار مورد محاسبه
۰,۵۸۳۹۴۱۶۰۵۸۳۹۴۱۶	دقت:
۰,۵۸۵۴۷۶۸۹۳۹۲۴۴۹۰۲	F1 Score(macro)

دقت و f1 بدست آمده چیزی نزدیک به ۶۰ درصد می‌باشند که باتوجه به نوع ماهیت وظیفه و

همچنین نوع شبکه‌ی استفاده شده تقریباً می‌توان بیان داشت که عملکرد به نسبت خوبی ارائه شده است. شاید بتوان کمی از عملکرد ضعیف بدست آمده را بدلیل شباهت‌هایی که بین برخی از آهنگ‌ها بوده است بیان کرد، که این شباهت‌ها سبب شده‌اند مدل به خوبی نتواند انواع مختلف موسیقی‌ها را تمایز دهد. به عنوان مثال در برخی از آهنگ‌های محلی خواننده به زبان محلی خود در یک سبک خاص موسیقایی آهنگ خود را خوانده است و در صورتی که یک خواننده دیگر با زبان محلی دیگر در همان سبک موسیقی آهنگی را خوانده باشد هر چند که نوع زبان خواننده متفاوت است. اما می‌تواند این موضوع در خطای مدل ما تاثیر گذار باشد. شاید بتوان بیان کرد به منظور کاهش این خطاها باید ویژگی‌هایی را صرفاً براساس صدای خواننده استخراج کرد تا بتواند تمایز بیشتری را در این وظیفه که دسته‌بندی موسیقی‌های محلی است ایجاد کند و دقت مدل افزایش یابد. همچنین حتی می‌توان با این داده‌ها توسط شبکه‌های پیچیده‌تر دیگری مانند CNN یا LSTM به دقت‌های نسبتاً بالاتری دست پیدا کرد. اما در پایان این بخش یک نمودار Confusion matrix براساس نتایج به دست آمده در شکل زیر رسم شده است.

همانطور که در این نمودار مشخص است، محور افقی تعداد آهنگ‌های پیش‌بینی شده در هر دسته را نشان می‌دهد و هر چه مقادیر موجود در قطر اصلی این ماتریس کم‌رنگ تر باشد نشان دهنده‌ی عملکرد مناسب مدل ایجاد شده در پیش‌بینی آهنگ‌ها می‌باشد.



شکل ۱۲: Confusion Matrix For MLP Classifier

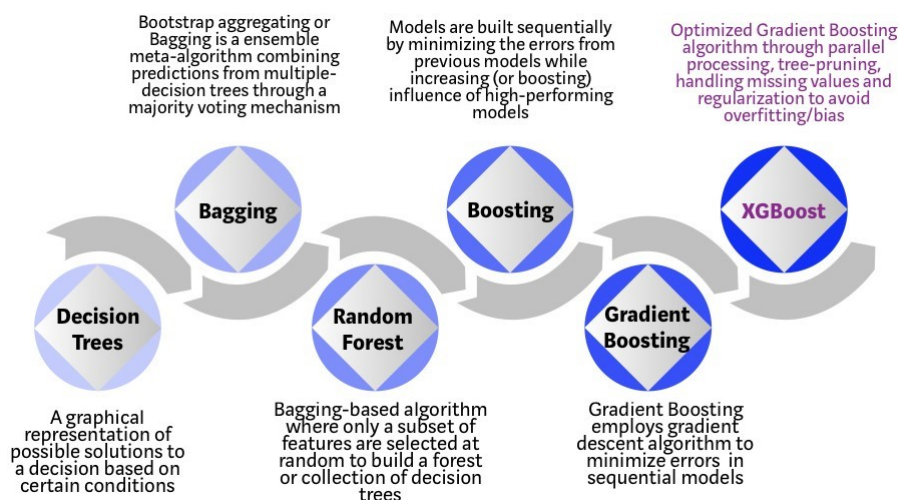
براساس این نتایج به وضوح مشخص است که، عملکرد مدل تقریباً در دسته‌بندی موسیقی‌ها مناسب بوده است و مدل بهترین عملکرد را در موسیقی ترکی داشته است و توانسته است بخش

زیادی از موسیقی‌ها این دسته را به درستی تشخیص دهد و عملکرد این مدل در تشخیص موسیقی لری به نسبت سایر موسیقی‌ها کمی نامناسب‌تر بوده است. به طور کلی می‌توان دلیل این ضعف را هم در ماهیت داده‌ها به دلیل شباهت برخی موسیقی‌ها (از لحاظ سبک موسیقایی، زبان خواننده و...)، شبکه‌ی عصبی استفاده شده (شبکه MLP) که مدل خیلی پیچیده‌ای براین وظیفه نیست و هم ویژگی‌های موجود که شاید براساس نوع وظیفه آنچنان مناسب آن نباشند بیان کرد.

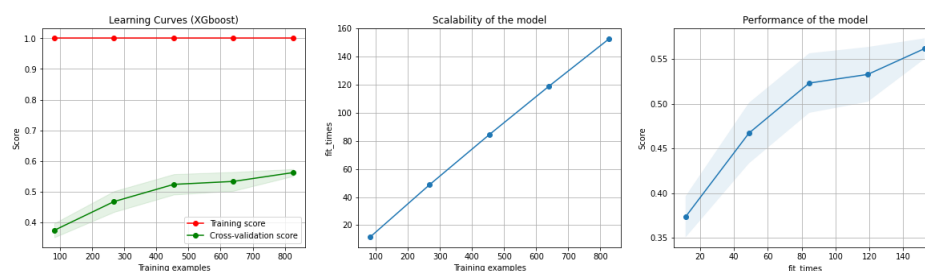
۲.۳ طبقه‌بندی به کمک شبکه XGBoost:

XGBoost یک طبقه‌بند از دسته‌ی ensemble learning است که به طور کلی از ترکیب چندین و چند مدل ضعیف‌تر و قرار دادن آن‌ها به صورت موازی یا سری به یک طبقه‌بند قوی‌تر می‌رسد. یک درخت تصمیم که واحدهای سازنده‌ی جنگل‌های تصادفی هستند در هر مرحله از عمقشان فضا را با استفاده از تنها یک ویژگی به دو قسمت تقسیم می‌کنند و با عمق بیشتر افزایش‌های بیشتر و قوانین طبقه‌بندی بیشتری را ایجاد می‌کنند. درخت‌های تصمیم به دلیل ساختار ساده‌ی خود معمولاً دقت پایینی دارند و نتیجه‌ی خوبی در دنیای واقعی کسب نمی‌کنند اما سازندگان الگوریتم ماشین‌لرنینگ با استفاده از اصل خرد جمعی (با رای‌گیری از تعداد افراد بسیار زیاد خطا به نسبت کم می‌شود) به این نتیجه رسیدند که هر بار درخت‌ها را با قسمت محدودی از ویژگی‌ها آموزش دهند و در نهایت از بین تمام آن‌ها رای‌گیری کنند. مشکل اصلی این الگوریتم این است که درخت‌ها از اشتباهات یکدیگر نمی‌آموزند و عملاً به درخت‌ها به صورت مستقل نگاه می‌شود. روش‌های boosting برای حل این مشکل به وجود آمدند. در روش‌های boosting درخت‌ها به جای این که در توازی یکدیگر باشند با یکدیگر سری هستند و هر درخت روی خطای درخت قبلی آموزش می‌بیند. روش‌های مبتنی بر گرادیان به این نحو عمل می‌کنند که درخت‌های بعدی هم‌جهت با تابع گرادیان ویژگی‌های قبلی حرکت می‌کنند. Xgboost یک پیاده‌سازی موازی از الگوریتم‌های مبتنی بر گرادیان است.

ما در طبقه‌بندی خود از روش xgboost به دلیل انعطاف بالای آن در جذب اطلاعات استفاده کردیم، که به دلیل تعداد داده‌های محدود پیش‌بینی می‌شد که مدل به داده‌های آموزش چسبیده شود که نتایج این نکته را تایید می‌کرد اما نکته‌ی بسیار قابل توجه این است که این چسبندگی باعث عدم عمومیت‌بخشی نشد و این الگوریتم از تمام الگوریتم‌های دیگر بر روی داده‌ی ارزیابی عملکرد بهتری داشت. عکس‌های زیر گفته‌های ما را تایید میکند، با توجه به اینکه Cross-Validation Score همواره با افزایش نمونه‌ها در حال افزایش است، میتوان نتیجه‌گیری کرد که مدل تشنه داده است. برای درک بهتر باید مد نظر داشته باشیم دو منبع خطا به طور کلی وجود دارد: الف) اشتباه در فرض اولیه برای مدل ب) کم بودن تعداد داده‌ها برای ابعاد مسئله حال با توجه به نمودارهای شکل ۱۴ و دقتی که برای مدل به دست آوردیم می‌توان نتیجه



شکل ۱۳: Evolution of XGBoost Algorithm from Decision Trees



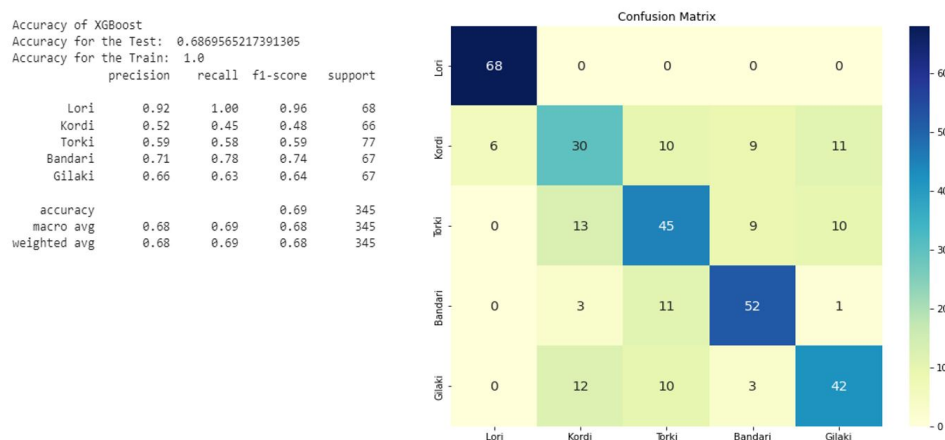
شکل ۱۴: Learning Curves, Scalability of Model, Performance of the model by fit times;

as we can see variation of model performance and Cross-Val Score has been shown.

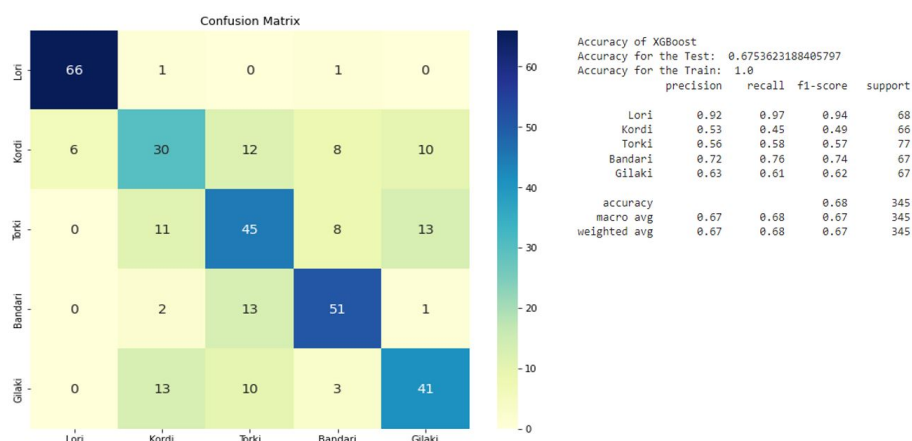
گرفت که در مدل ساخته خطای "ب" جلوگیری میکند از عملکرد بهتر مدل. در نهایت عملکرد مدل را نیز میتوان مشاهده کرد به ازای فیچر های مختلف:

با توجه نتایج مشاهده میکنیم که در دو کلاس لری و بندری با توجه به ریتیمی که دارند خطای کمتری وجود دارد اما در کلاس کردی، با توجه به این نکته که آهنگ های کردی از لحاظ ریتم و موسیقی همه طیف ها را در خود دارد در نتیجه بیشترین خطا را دارد. اما در مورد خطا متوجه میشویم که به داده های آموزش over fit شده است و برای حل مشکل دو راه وجود دارد، استفاده از مدل ساده تر و افزایش نمونه ها. در نمودار cross val score و توضیحات آن گفته شد که اضافه کردن داده به مدل میتواند راه حل مناسب تری باشد.

همانطور که در بخش انتخاب فیچر ها گفته شد، انتخاب ویژگی ها و کاهش آن ها باعث عملکرد مناسب مدل میشود اما به طور کلی باعث افزایش دقت نشد چرا که میزان سیگنال به نویز مقدار کمی است. اگر XGboost را به ویژگی های به دست آمده با PCA اعمال کنیم همانطور که گفته



شکل ۱۵: Accuracy of XGboost for Raw Data and Confusion Matrix



شکل ۱۶: XGBosst With Feature selection (Wrapper Methods) Results

شد نتایج گرفته شده بسیار بد می باشد و فیچر های اشتباه به عنوان Principal Component انتخاب میشوند.

تنها نکته امیدوار کننده راجب این طبقه بند کاهش چسبندگی آن به داده های آموزش است اما با توجه به دقت داده های تست خروجی ها اصلا قابل قبول نیستند.

۳.۳ طبقه بندی به کمک شبکه SVM:

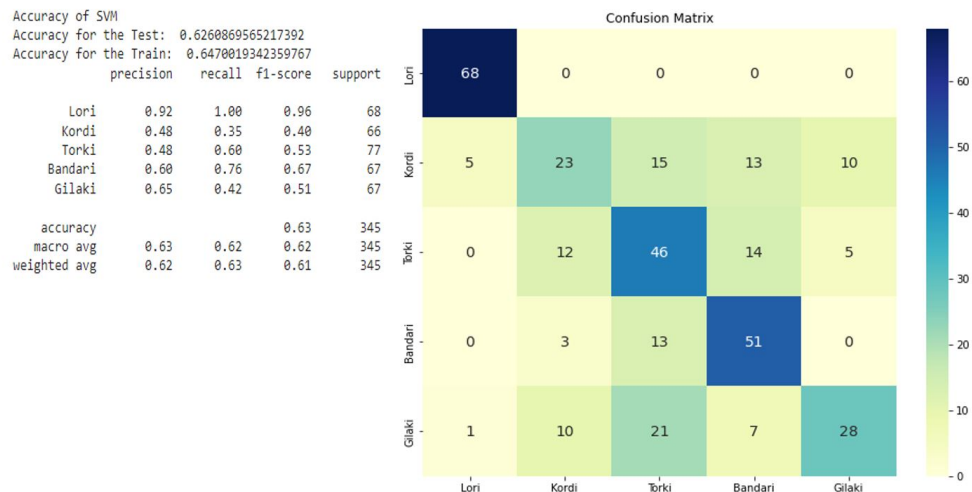
مدل آخری که استفاده شده است SVM است که نتایج آن در این قسمت آمده است. با استانداردیز کردن داده ها و سپس اعمال SVM، میتوان نمودار های یادگیری (learning curve) (میتوان و Cross Val Score را مشاهده کرد. با توجه به اینکه بعد از ۶۰۰ نمونه آموزشی Score

Accuracy for the Test: 0.2985507246376812
 Accuracy for the Train: 0.9574468085106383

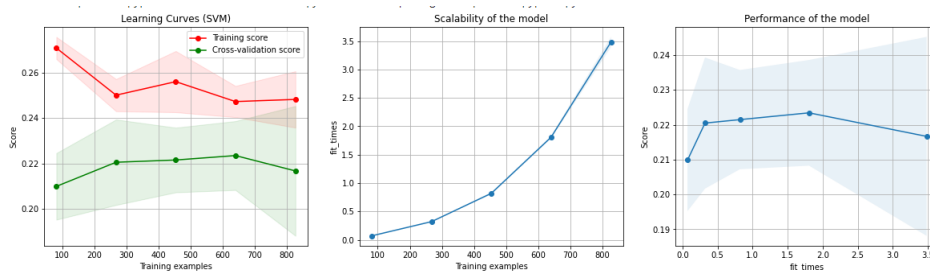
	precision	recall	f1-score	support
Lori	0.32	0.97	0.49	68
Kordi	0.28	0.39	0.33	66
Torki	0.22	0.08	0.12	77
Bandari	0.31	0.06	0.10	67
Gilaki	0.14	0.01	0.03	67
accuracy			0.30	345
macro avg	0.25	0.30	0.21	345
weighted avg	0.25	0.30	0.21	345

شکل ۱۷: Report for XGBoost Classifier after Performing PCA on data

کاهش یافته است میتوان نتیجه گرفت که اگر نمونه ها زیاد شود دیگر عملکرد مدل بهتر نمی شود. به عبارت دیگر مدل بایاس دارد و به اندازه کافی Variation داده ها را نمیتواند پوشش دهد.



شکل ۱۸: Confusion and report For SVM on Standardized Features



شکل ۱۹: Learning Curves For SVM

با اعمال PCA همانطور که انتظار داشتیم و قبلا توضیح داده شد عملکرد کلی مدل کاهش

Accuracy of SVM with Selected Features					Accuracy of SVM with PCA				
Accuracy for the Test: 0.6753623188405797					Accuracy for the Test: 0.4927536231884058				
Accuracy for the Train: 1.0					Accuracy for the Train: 0.6005802707930368				
	precision	recall	f1-score	support		precision	recall	f1-score	support
Lori	0.92	0.97	0.94	68	Lori	0.91	0.99	0.94	68
Kordi	0.53	0.45	0.49	66	Kordi	0.31	0.32	0.31	66
Torki	0.56	0.58	0.57	77	Torki	0.41	0.58	0.48	77
Bandari	0.72	0.76	0.74	67	Bandari	0.41	0.39	0.40	67
Gilaki	0.63	0.61	0.62	67	Gilaki	0.38	0.16	0.23	67
accuracy			0.68	345	accuracy			0.49	345
macro avg	0.67	0.68	0.67	345	macro avg	0.48	0.49	0.47	345
weighted avg	0.67	0.68	0.67	345	weighted avg	0.48	0.49	0.47	345

Report For SVM classifier applied to Features Extracted From SFS(Left) and شکل ۲۰: PCA(right)

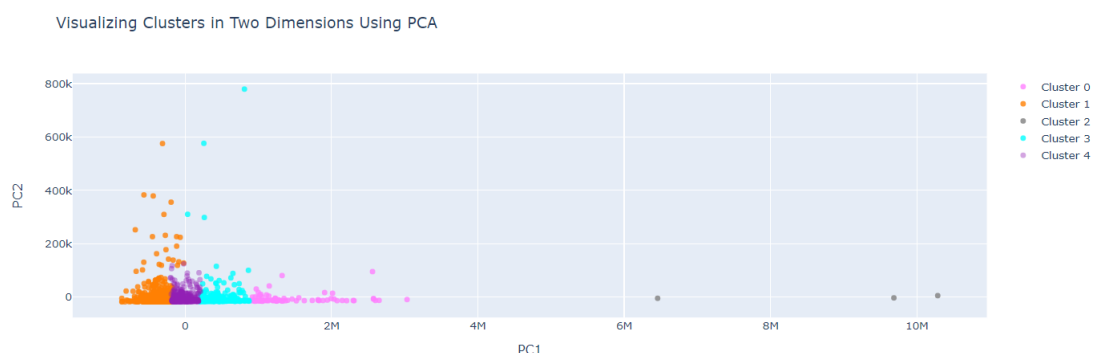
یافت و با اعمال SFS میتوان مشاهده کرد که عملکرد مدل بهتر شده است اما دچار Over fitting شده است که قابل حدس بود چرا که Objective Function ما در ابتدا برای انتخاب ویژگی ها با SFS از مدل خطی استفاده کردیم (مشابه SVM است) در نتیجه به شدت ویژگی های ما کمبود generality دارند در نتیجه overfit میشود.

۴ خوشه‌بندی

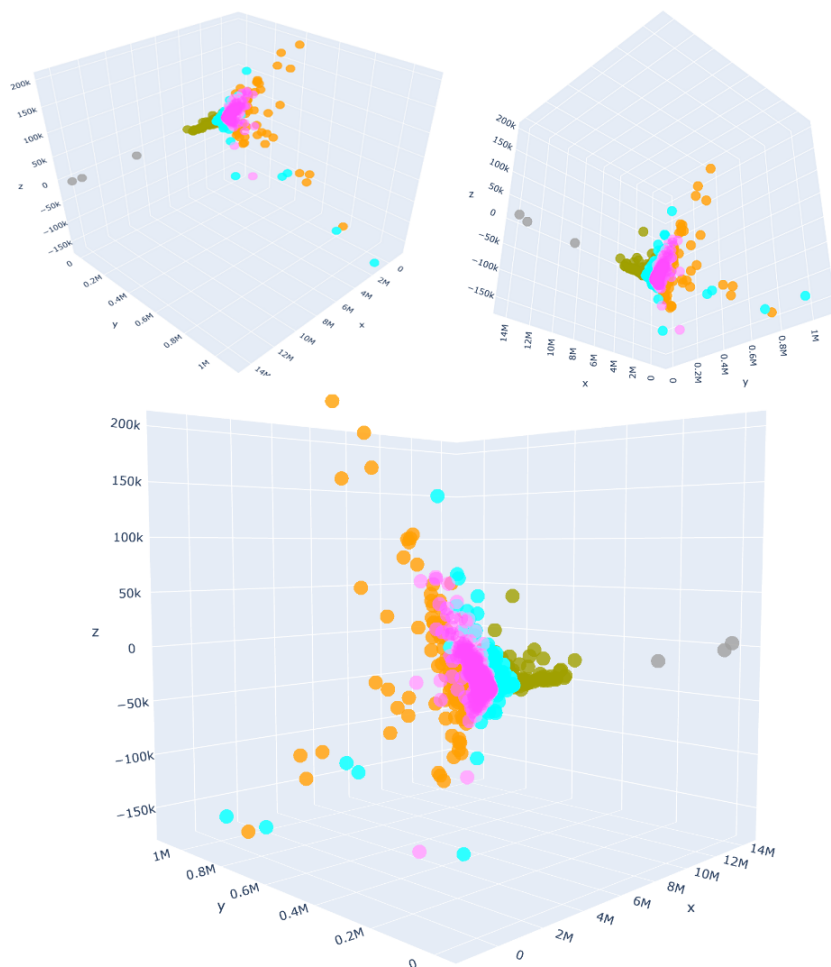
در این بخش از پروژه، مطابق صورت سوال از ما خواسته شده است که دو الگوریتم مختلف خوشه‌بندی را به ازای سائز خوشه‌های یک تا پنج بر روی داده‌های ورودی اعمال کنیم. در ابتدای این بخش به منظور این که یک دید کلی نسبت به داده‌ها داشته باشیم، داده‌ها را در قالب یک نمودار سه‌بعدی نمایش می‌دهیم. در قدم بعدی داده‌ها را با استفاده از الگوریتم‌های خوشه‌بندی K-means و الگوریتم خوشه‌بندی سلسله مراتبی خوشه‌بندی می‌کنیم. در هنگام اعمال الگوریتم K-means با استفاده از در یک نمودار تعداد بهینه خوشه‌ها نمایش می‌دهیم، سپس به ازای هر کدام از خوشه‌های بدست آمده معیارهای Purity، Rand Index و F1 را محاسبه می‌کنیم و در آخر تحلیلی بر روی نتایج بدست می‌آوریم.

۱.۴ استفاده از روش PCA برای کاهش ابعاد:

ابعاد داده بسیار بالا است و درک آن برای ما بسیار سخت است برای این که درک اولیه‌ای از خوشه‌ها داشته باشیم سعی می‌کنیم که آن‌ها را در سه بعد نشان دهیم برای کاهش ابعاد از Principle Component Analysis استفاده می‌کنیم. می‌دانیم که PCA روشی است که محورهایی را انتخاب می‌کند که بیشترین واریانس را در خود ذخیره کنند. در نتیجه با اعمال PCA و کاهش ابعاد به ۳ به شکل زیر می‌رسیم. توجه کنید که خوشه‌بندی در همان ابعاد اصلی اتفاق افتاده است و صرفاً برای مصورسازی به ابعاد پایین‌تر منتقل شده است.



شکل ۲۱: Using PCA for visualizing clustering output in 2 dimensions.

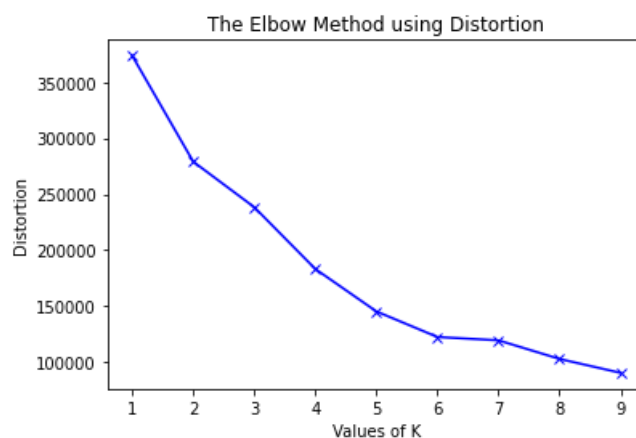


شکل ۲۲: Visulizing by PCA after clustering.

۲.۴ خوشه‌بندی به روش k نزدیکترین مرکز:

در این قسمت، در ابتدا الگوریتم K-means را به عنوان الگوریتم خوشه‌بندی در نظر می‌گیریم و سایر خوشه‌ها را از ۱ تا ۵ مطابق خواسته‌ی مسئله تغییر می‌دهیم. همچنین ما نمودار مربوط به میزان Distortion (میانگین مجذور فواصل از مراکز خوشه‌ها) به کمک فاصله اقلیدسی (براساس تعداد خوشه‌های یک تا ۱۰ را برای این الگوریتم رسم می‌کنیم که نتایج مربوط به آن مطابق نمودار زیر می‌شود.

در این نمودار محور افق تعداد خوشه‌ها و محور عمود نشان‌دهنده‌ی میزان Distortion می‌باشد همانطور که در این نمودار مشخص است، نقاطی که در آن‌ها زانوهایی رخ داده است، نشان دهنده‌ی



شکل ۲۳: The Elbow Method using Distortion

نقاط مناسب برای تعداد خوشه‌ها می‌باشد، و همانطور که مشاهده می‌شود در اندازه‌ی خوشه‌ی دو و هفت این زانوها اتفاق افتاده است. (شایان ذکر است به منظور داشتن دید کامل‌تر ما تعداد خوشه‌ها را در این بخش تا ۱۰ افزایش دادیم اما در تحلیل نتایج تعداد خوشه‌ها را یک تا پنج در نظر گرفتیم)

همچنین ما به منظور ارزیابی نتایج بدست آمده از معیارهای زیر به ازای هر کدام از اندازه‌های خوشه‌ی یک تا پنج محاسبه می‌کنیم. معیار اول) شاخص خلوص (Purity) است که درصد مطابقت بین برچسب‌های خوشه‌بندی و برچسب‌های واقعی را می‌سنجد (این معیار هر چه به یک نزدیکتر باشد نشان دهنده‌ی کیفیت مناسب خوشه‌بندی است).

معیار دوم) شاخص رند اصلاح شده (Rand Index) است که میزان شباهت بین دو شیوه‌ی برچسب گذاری را می‌توان با این شاخص اندازه گیری کرد (این معیار هر چه به یک نزدیکتر باشد نشان دهنده‌ی کیفیت مناسب خوشه‌بندی است).

معیار سوم) F1 score که به این منظور باید مقادیر مثبت صحیح، مثبت کاذب، منفی صحیح و منفی کاذب در این وظیفه‌ی خوشه‌بندی محاسبه شود که ما براساس تعاریفی که برای مقادیر مثبت صحیح، مثبت کاذب و ... در مسئله خوشه‌بندی وجود دارد (در ادامه تعاریف بیان شده است) یک تابع نوشتیم و نتایج را با آن‌ها ارزیابی کردیم.

منظور از مثبت صحیح، آن زوج مشاهداتی هستند که در یک دسته هستند و در یک خوشه نیز قرار گرفته‌اند.

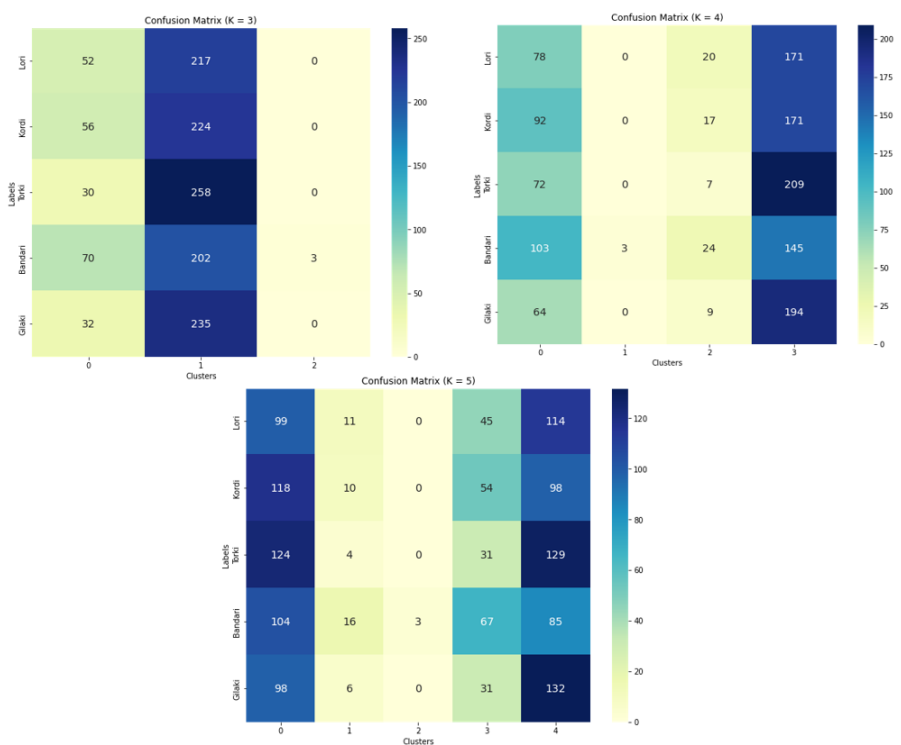
منظور از منفی صحیح، آن زوج مشاهداتی هستند که در دو دسته مجزا هستند و در خوشه مجزا نیز قرار گرفته اند.

مثبت کاذب، آن زوج مشاهداتی است که در دو دسته مجزا قرار گرفته‌اند و در یک خوشه قرار دارند.

منفی کاذب، آن زوج مشاهداتی که در یک دسته قرار دارند و به اشتباه در دو خوشه قرار گرفته‌اند. اکنون تمامی این معیارها را به ازای اندازه‌ی خوشه‌های یک تا پنج براساس الگوریتم K-means نمایش می‌دهیم در قالب یک جدول نشان می‌دهیم.

F1-score	Rand Index	Purity	
0.332739	0.199572	0.208847	1
0.323365	0.302613	0.227701	2
0.314203	0.377106	0.240029	3
0.355945	0.498696	0.245830	4
0.261041	0.594565	0.248006	5

در ادامه نیز به ازای اندازه‌ی خوشه‌های ۳، ۴ و ۵ ماتریس درهم‌ریختگی رسم می‌شود تا تعداد اهنگ‌های قرار گرفته شده در هر خوشه نمایش داده شود.



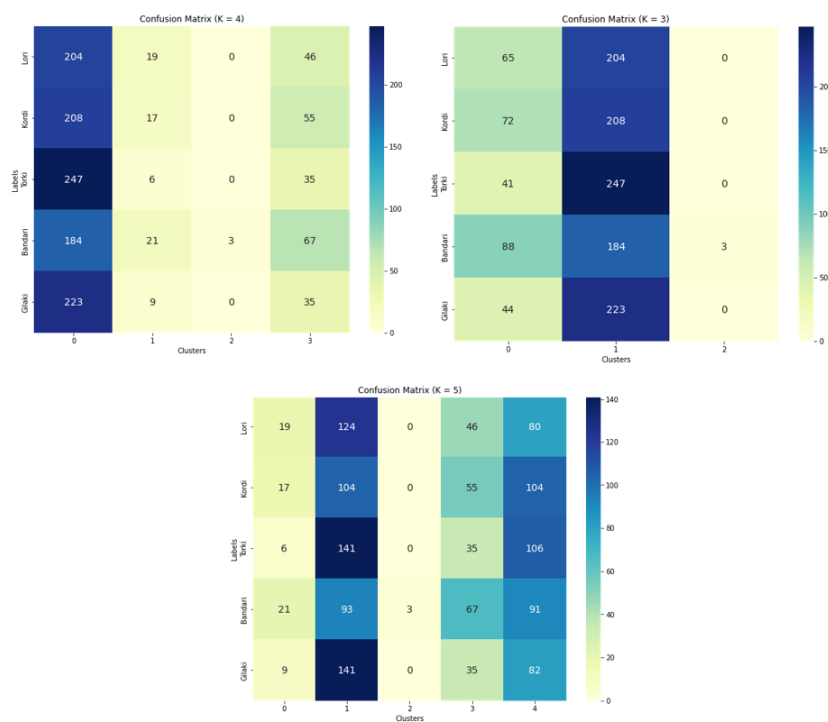
شکل ۲۴: Confusion matrix of K-means clustering

۳.۴ خوشه‌بندی به روش سلسله مراتبی :

در این بخش نیز نتایج الگوریتم سلسله مراتبی به ازای سایز خوشه‌های یک تا پنج بررسی می‌شود. در جدول زیر معیارها مورد استفاده به ازای هر اندازه‌ی خوشه محاسبه شده است و نمایش داده می‌شود.

F1-score	Rand Index	Purity	
0.332739	0.199572	0.208847	1
0.309396	0.414090	0.245105	2
0.309098	0.414513	0.245105	3
0.309112	0.424674	0.245105	4
0.297711	0.601255	0.245105	5

در نمودارهای زیر نیز ماتریس درهم‌ریختگی به ازای سایز خوشه‌های ۳ تا ۵ نمایش داده می‌شود که تعداد هر دسته از اهنگ‌های قرار گرفته شده در هر خوشه را نمایش می‌دهد.



شکل ۲۵: Confusion matrix of Hieratical clustering

اکنون لازم است که پس از نمایش نمودارها و مقادیر بدست آمده برای معیارهای مختلف، به تحلیل این معیارها بپردازیم. همانطور که از نتایج هر دو الگوریتم مشخص است، تقریباً نتایج بدست

آمده برای معیارهای در نظر گرفته شده برای هر دو الگوریتم با یکدیگر برابر هستند. اما معیار Rand Index به ازای اندازه‌ی خوشه‌ی ۵ در الگوریتم سلسله مراتبی کمی از الگوریتم K-means بهتر عمل کرده است و مقدار حدود ۶۰ درصد بدست آمده است. براساس نتایج بدست آمده، به وضوح مشخص است این الگوریتم‌ها نتوانسته‌اند، به خوبی عمل خوشه‌بندی هر دسته از آهنگ‌ها را انجام بدهند و هر کدام از آهنگ‌های محلی را در خوشه‌های جداگانه‌ای قرار بدهند. این نتایج به وضوح از ماتریس‌های آشفتگی مربوط به ازای سائز خوشه‌ی ۵ مشخص است. در هر دو این الگوریتم‌ها ۵ دسته آهنگ بیشتر در بین ۳ خوشه قرار گرفته‌اند. با توجه به مشاهده‌ی ماتریس آشفتگی مربوط به الگوریتم سلسله مراتبی با اندازه‌ی خوشه‌ی ۵ در می‌یابیم که ترکی و گیلکی بسیار با یکدیگر شبیه هستند چرا که تمپوی آن‌ها مشابه و از طرفی واج‌هایی که در آهنگ‌ها استفاده می‌شود بسیار به یکدیگر مشابه هستند و همچنین نزدیکی اقلیمی این دو قومیت نیز باعث شده است که از لحاظ موسیقایی آهنگ‌های مشابهی داشته باشند. همچنین با مشاهده ماتریس آشفتگی مربوط به الگوریتم K-means با ۵ عدد خوشه، بازهم مشاهده می‌شود که داده‌ها تنها در سه خوشه قرار گرفته‌اند به منظور علت این موضوع باید به خود داده‌ها رجوع کنیم اگر توجه کنیم می‌فهمیم که برای مثال آهنگ‌های کردی صرفاً زبان متفاوتی دارند اما ریتم، ملودی، تمپو و ویژگی‌های موسیقایی‌شان بسیار مشابه کلاس‌های دیگر است و این نوع داده در تمام فضای ویژگی پخش است به بیان دیگر پراکندگی (variation) زیادی دارد پس باید انتظار همین را می‌داشت که الگوریتم نتواند تفاوت بین آن و باقی کلاس‌ها قائل شود.

از دیدگاهی دیگر مدل ایجاد شده دارای بایاس است چرا که تنها سه برجسب را تشخیص می‌دهد در نتیجه برای بهبود مدل باید از روش‌های پیچیده‌تر مانند DBscan که فرض شکل به خصوصی برای داده نمی‌کنند را استفاده کنیم. دلیل عدم آوردن نتایج DBscan در گزارش این است که الگوریتم DBscan بسیار بدتر در مقیاس عمل می‌کند و داده‌های ما بسیار ابعاد بالایی دارند.

۵ ضمیمه: نحوه اجرای برنامه و مشاهده خروجی

الف) برای استخراج ویژگی ها باید فایل "feature-extraction" را اجرا کرد و نیاز به اتصال به اینترنت است، توصیه میشود از کولب یا Kaggle برای اجرا استفاده شود.

ب) برای مشاهده عکس های قسمت استخراج ویژگی ها و پیش پردازش و انتخاب آنان کافیست برنامه Feature-Visulization-01 اجرا شود و آهنگ شماره ۲ در پوشه قرار بگیرد.

ج) برای مشاهده خروجی XGBoost و SVM فایل XGBost-SVM-Classifiers-02 اجرا شود و خروجی قسمت الف در پوشه قرار بگیرد.

د) برای قسمت خوشه بندی دو فایل Clustering-I و Clustering-II قرار داده شده است. برای اجرا مانند قسمت ج نیاز است خروجی قسمت الف در پوشه مناسب قرار بگیرد و سپس اجرا شوند. نکته دیگر اینکه خروجی فایل Clustering-II در فایل قابل مشاهده نیست چرا که دارای اشکال سه بعدی بوده است، برای دسترسی به خروجی از لینک زیر استفاده شود.

<https://www.kaggle.com/hamedgholamy/clustering>

مراجع

Surfboard: Audio Feature Extraction for Modern Machine Learning Raphael [١]

Lenain, Jack Weston, Abhishek Shivkumar, Emil Fristed

Novel techniques for Audio Music Classification and Search Kris West [٢]

Ankit Pal and Malaikannan Sankarasubbu. 2021. Pay attention to the cough: [٣]

early diagnosis of COVID-19 using interpretable symptoms embeddings with
cough sound signal processing. Proceedings of the 36th Annual ACM Sympos-
sium on Applied Computing. Association for Computing Machinery, New York,

NY, USA, 620–628

Song-Level Features and Support Vector Machines for Music Classification Man- [٤]

del, Michael I.; Ellis, Daniel P. W.