

دانشگاه تهران

دانشکده فنی

گروه برق و کامپیوتر



گزارش اولیه پروژه پایانی یادگیری ماشین

اساتید درس:

دکتر نجار اعرابی

دکتر ابوالقاسمی دهاقانی

دکتر توسلی پور

گردآورندگان:

محمد سجاد علیدار ۸۱۰۱۰۱۲۳۱

بهزاد موسایی شیرمحمد ۸۱۰۱۰۱۲۷۸

میلیکا فتحی پور ۸۱۰۱۰۰۴۲۴

محمد جواد رنجبر کلهروودی ۸۱۰۱۰۱۱۷۳

زمستان ۱۴۰۱

فهرست مطالب

۷	(۰) پیوست
۷	(۱) مقدمه
۹	(۲) ویژگی های موسیقی
۹	(۱-۱) موسیقی چیست؟
۹	(۲-۱) فرمت های صوتی
۱۰	(۳-۱) سیگنال صوتی
۱۰	(۴-۱) ویژگی های حوزه زمان
۱۲	(۵-۱) ویژگی های حوزه فرکانس
۱۸	(۶-۱) ویژگی های طیفی
۲۰	(۳) استخراج ویژگی و پیش پردازش داده ها
۲۰	(۱-۳) دیتاست
۲۱	(۲-۳) استخراج ویژگی
۲۵	(۳-۳) پیش پردازش داده ها
۲۷	(۴-۳) کاهش بعد
۳۰	(۵-۳) انتخاب ویژگی
۳۰	(۴) طبقه بندی کننده ها
۳۰	(۱-۴) شبکه عصبی چندلایه
۳۲	SVM (۲-۴)
۳۹	Knn (۳-۴)
۴۲	Logistic Regression (۴-۴)
۴۳	Ensemble Learning (۵-۴)
۴۵	XGboost (۶-۴)
۴۷	مقایسه (۷-۴)
۴۷	(۵) خوشه بندی
۴۸	K-means (۱-۵)
۵۰	Spectral Clustering (۲-۵)

۶) نتیجه گیری

۷) ضمیمه

فهرست تصاویر

۹	شمای کلی سیگنال موسیقی	۱
۱۰	نمایش یک سیگنال صوتی	۲
۱۱	نمونه‌ای از سیگنال موسیقی در دستگاه نوا و شور	۳
۱۱	نرخ گذر از صفر	۴
۱۳	اسپکتروگرام یک موسیقی	۵
۱۳	نمونه‌ای از SFTF	۶
۱۴	نمودارهای فرکانس-شدت	۷
۱۵	ویژگی‌های کروما	۸
۱۵	مراحل آنالیز کپستروم	۹
۱۶	شمایی از Harmonics and Perceptual	۱۰
۱۷	نمونه از نمایش تعداد ضربات در دقیقه	۱۱
۱۸	نمایی از Mel-Frequency Cepstral	۱۲
۱۹	نمایی از Spectrogram	۱۳
۲۰	نمایی از Mel Spectrogram	۱۴
۲۱	نمایی از Spectral Centroid	۱۵
۲۲	نمایی از Spectral Bandwidth	۱۶
۲۲	نمونه‌ای از کنتراست طیغی	۱۷
۲۳	نمونه‌ای از طیف رولوف	۱۸
۲۴	نمایی از Spectral Rolloff	۱۹
۲۵	ویژگی‌های دیتاست	۲۰
۲۶	ماتریس همبستگی	۲۱
۲۷	توزیع داده‌ها	۲۲
۲۸	پیش‌پردازش داده	۲۳
۲۸	PCA	۲۴
۲۹	LDA	۲۵
۳۱	مدل MLP	۲۶
۳۲	ماتریس پراکندگی MLP	۲۷
۳۳	کرنل‌های مختلف SVM	۲۸
۳۹	ماتریس پراکندگی svm‌ها	۲۹

۴۰	knn	۳۰
۴۱	knn ماتریس پراکندگی مدل	۳۱
۴۲	ماتریس پراکندگی Logistic Regression	۳۲
۴۴	نتایج Ensemble Learning بر روی داده های train	۳۳
۴۶	نتایج Ensemble Learning بر روی داده های test	۳۴
۴۷	نتایج Ensemble Learning بر روی داده های غیر نرمال train	۳۵
۴۹	نتایج Ensemble Learning بر روی داده های غیر نرمال test	۳۶
۵۰	نتایج XGboost بر روی داده های غیر نرمال train	۳۷
۵۲	نتایج XGboost بر روی داده های غیر نرمال test	۳۸
۵۲	دسته بندی های مختلف با kmeans	۳۹
۵۳	دسته بندی های مختلف با kmeans	۴۰
۵۳	دسته بندی های مختلف با kmeans	۴۱
۵۴	خوشه بندی K-means	۴۲
۵۴	نمودار زانویی K-means	۴۳
۵۵	ماتریس پاکندگی spectral	۴۴
۵۶	ماتریس پاکندگی spectral	۴۵
۵۷	ماتریس پاکندگی spectral	۴۶
۵۷	خوشه بندی spectral	۴۷
۵۸	خوشه بندی spectral	۴۸
۵۸	خوشه بندی spectral	۴۹
۵۸	نرمالایز شده و نشده	۵۰
۵۹	svm برای LDA	۵۱
۵۹	svm برای PCA	۵۲

(۰) پیوست

تمامی موارد خواسته شده در پروژه به طور کامل درون فایل notebook است همچنین تمامی موارد خواسته شده در گزارش توضیه داده شده اما برای جلوگیری از طولانی شدن تعدادی محدود از جدول‌ها و شکل‌ها برای تحلیل آورده شده است.

(۱) مقدمه

در این پروژه قصد داریم مدل‌هایی برای تشخیص دستگاه موسیقی ایرانی پیاده سازی کنیم. این مدل‌ها با استفاده از الگوریتم‌های یادگیری ماشین این دستگاه‌ها را طبقه‌بندی و خوشبندی می‌کنند. برای این منظور مراحل مختلفی را باید طی کنیم تا داده‌ها را آماده برای انجام این کار کنیم. خلاصه‌ای از مراحل در ادامه توضیح داده شده است:

جمع آوری داده:

ابتدا باید مجموعه داده‌ای^۱ برای این کار آماده کنیم، این مجموعه داده توسط دانشجویان دانشگاه تهران به روش جمع‌سپاری^۲ شده است. این داده‌ها به فرمت mp3 در ۷ کلاس شور، سه‌گاه، ماهور، همایون، راست‌پنجم‌گاه، نوا و چهارگاه هستند. طول این قطعات موسیقی بین ۲۰ تا ۶۰ ثانیه می‌باشد. همینطور، سازهایی که این قطعات در آن‌ها نواخته شده‌اند نیز در داخل این مجموعه داده مشخص شده است. استخراج ویژگی:

موسیقی نمونه‌ای از سیگنال صوتی می‌باشد که ویژگی‌های مخصوص خود را دارا می‌باشد. ما برای آموزش مدل مناسب که قابلیت تعیین‌پذیری داشته باشد باید، ویژگی‌هایی که در مشخص کردن دستگاه موسیقی اثر دارد را پیدا کرده و استخراج کنیم. مصورسازی و تمیز کردن داده‌ها:

برای اینکه مدلی مناسب برای این کار درست کنیم، بسیار اهمیت دارد که داده‌های آموزش کمترین مقدار خطأ را داشته باشد تا از آموزش داده شدن مدل با داده‌های نویزی^۳ جلوگیری کنیم. بدین منظور بایست ابتدا داده‌ها را بر اساس ویژگی‌ها مصورسازی کرده تا هم در صورت وجود داده‌های نادرست آن‌ها را بیاییم و حذف کنیم و همینطور با بررسی الگوهای داده بهترین مدل را برای وظیفه طبقه‌بندی انتخاب کنیم. همچنین، در صورت نیاز می‌توان داده‌های موسیقی را به چندین داده با طول کمتر تبدیل کرد تا مجموعه داده‌ای با تعداد داده‌ی بیشتر دست باییم.

تصورسازی و تمیز کردن داده‌ها طبقه‌بندی:

حال پس از تمیز کردن داده و استخراج ویژگی‌های آن، می‌توانیم بر اساس اطلاعات به دست آمده از گام قبلی مدل‌هایی طراحی کنیم که بهترین عملکرد را برای داده‌ی ما بدهنند. در این پروژه از مدل‌های طبقه‌بند مانند SVM، شبکه عصبی چند لایه^۴ و XGBoost برای این کار استفاده می‌کنیم.

خوشبندی:

در این مرحله نیز با استفاده از روش‌های خوشبندی و اطلاعات به دست آمده در گام‌های قبلی، سعی در جداسازی داده‌ها خواهیم کرد.

Dataset^۱
Crowdsourcing^۲
Noise^۳
perceptron Multilayer^۴

در این پروژه از دو مدل Agglomerative و Kmeans برای جداسازی داده‌ها استفاده خواهد شد.
در ادامه تمامی مراحل بالا را به صورت کامل پیاده‌سازی کرده و توضیح می‌دهیم.

(۲) ویژگی‌های موسیقی

(۱-۲) موسیقی چیست؟

موسیقی یک نمونه از سیگنال صوتی می‌باشد و در نتیجه تمامی ویژگی‌ها مربوط به یک سیگنال را دارا است. و به عبارتی همین ویژگی‌ها هستند که تفاوت را در قطعه‌های موسیقی ایجاد می‌کند.

به طور کلی سیگنال‌ها داری سه ویژگی معروف و مهم هستند که به شرح زیر مطرح می‌شوند:

(۱) دامنه: مشخص کننده این که چه مقدار سیگنال قوی و ضعیف هست.

(۲) فرکانس: به صورت ساده شده می‌توان گفت، تعداد دفعاتی که در واحد زمان، موج سیگنال تکرار می‌شود را فرکانس می‌گوییم. در سیگنال صوتی هر چقدر فرکانس افزایش یابد صدا زیرتر می‌شود و هر چقدر فرکانس کاهش یابد صدا بهتر می‌شود.

(۳) فاز: زاویه اولیه سیگنال در نقطه مبدأ است. (البته نسبت به دو مورد دیگر کاربرد کمتری در بحث ما دارد.)



شکل ۱: شمای کلی سیگنال موسیقی

(۲-۲) فرمت‌های صوتی

فایل‌های صوتی دیجیتال با فرمت WAV امواج صوتی هستند که با نمونه‌برداری از موسیقی آنالوگ در فواصل گسسته دیجیتالی می‌شوند. (معمولًاً ۱۰۴۴ کیلوهرتز برای صدای با کیفیت CD به این معنی که نمونه‌ها ۴۴۱۰۰ بار در ثانیه گرفته می‌شوند). هر نمونه، برابر با مقدار دامنه سیگنال در یک بازه زمانی خاص است. نکته اصلی در مورد فرمت WAV این است که یک فرمت بدون فشرده‌سازی یا Uncompressed است. به همین سبب کیفیت بالا و اصلی موسیقی آنالوگ را حفظ می‌کند ولی در مقابل حجم آن بسیار زیاد است. همچنین از ان جهت که این فرمت به صورت ۳۲ بیت بدون علامت تعریف می‌شود حداقل تا ۴ گیگابیت می‌تواند حجم داشته باشد بنابراین مدت زمان ضبط موسیقی با این فرمت، نسبت به فرمت‌های فشرده مثل MP3 و MP4 و OGG محدود تر است.

در این پروژه فایل‌های صوتی به صورت MP3 تهیه شدند که در مورد این فرمت صوتی می‌توان گفت که یک فرمت رمزگذاری داده‌های صوتی دیجیتالی است که با استفاده از روش "اتلاف (حذف) داده‌ها" (Lossy data) این داده‌های صوتی را فشرده می‌کند؛ یعنی فرمت MP3 قسمت‌های خاصی از فایل صوتی اصلی را از بین می‌برد و سایر قسمت‌ها را فشرده می‌کند. اعمال کلی که در پروسه فشرده‌سازی

فایل صوتی توسط فرمت MP3 انجام می‌شود، عبارت اند از:

- حذف قسمت‌هایی از فایل صوتی که خارج از محدوده شنیداری گوش انسان است.

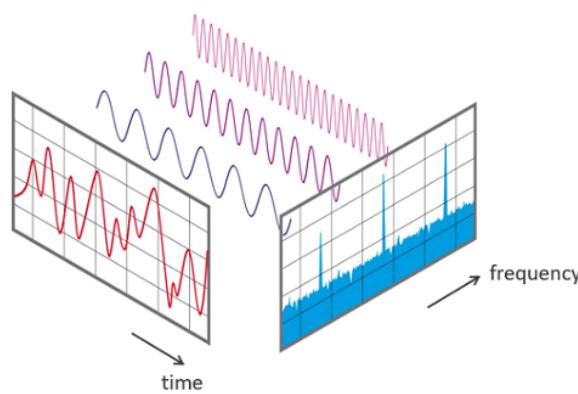
- حفظ لایه‌های صوتی‌ای که گوش انسان بهتر از سایر لایه‌های صوتی می‌شوند.

- حذف لایه‌های صوتی کوتاه تر که به صورت همزمان با لایه‌های صوتی بلندتر پخش می‌شوند.

لازم به ذکر است که کتابخانه‌های معروف کار با صوت در پایتون^۵ معمولاً با فایل‌های WAV کار می‌کنند لذا در این پروژه نیاز به تبدیل این فایل‌های mp3 به این فرمت داشتیم.

(۳-۲) سیگنال صوتی

بازگشته به مطلب بالا به طورکلی می‌توان به این نتیجه رسید که موسیقی به شکل یک سیگنال صوتی با پارامترهایی مانند فرکانس، پهنه‌باند و غیره نمایش داده می‌شود. یک سیگنال صوتی معمولی را می‌توان به عنوان تابعی از دامنه و زمان نشان داد که در شکل زیر مشخص است.



شکل ۲: نمایش یک سیگنال صوتی

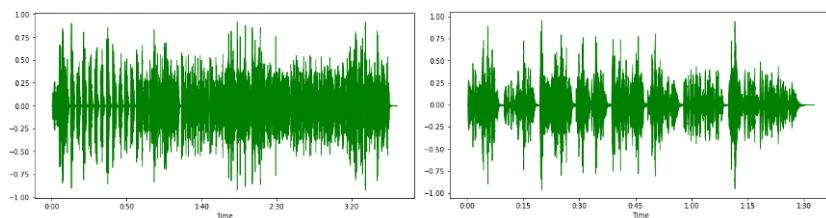
معمولًا این سیگنال‌های صوتی به فرمات‌های mp3 (MPEG-1 Audio Layer 3)، wav (Waveform Audio File) و WMA (Windows Media Audio) به صورت دیجیتالی موجود هستند. به عنوان مثال نمونه‌ای از سیگنال یک موسیقی در دو دستگاه نوا و شور را در ادامه مشاهده می‌کنید: حال با توجه به اطلاعاتی که مورد سیگنال‌های صوتی به دست آورده‌یم به بررسی ویژگی‌های اصلی و مورد استفاده در پروژه می‌پردازیم.

(۴-۲) ویژگی‌های حوزه زمان

نرخ عبور از صفر (Zero crossing rate).

نرخ عبور از صفر (ZCR) در یک سیگنال به معنی تعداد دفعاتی است که سیگنال از مثبت به صفر و به منفی یا از منفی به صفر و به مثبت

⁵ Python



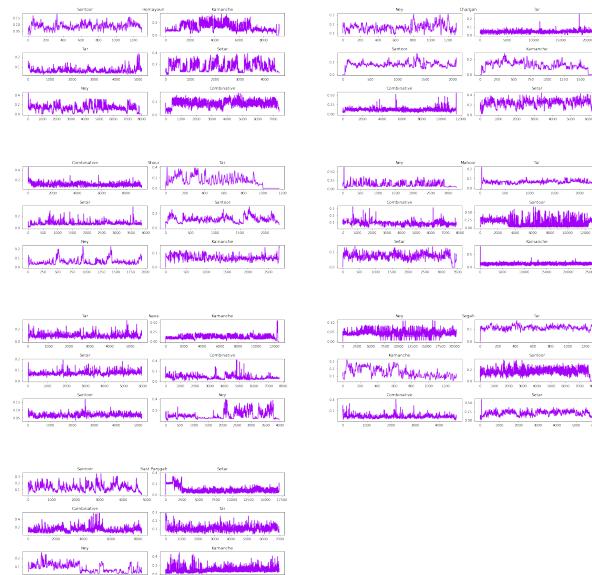
شکل ۳: نمونه‌ای از سیگنال موسیقی در دستگاه نوا و شور

تغییر می کند. یا به عبارت دیگر نرخ تغییر علامت سیگنال است. این ویژگی را می توان به صورت زیر نیز فرموله کرد:

$$ZCR = \frac{1}{T-1} \cdot \sum_{i=1}^{T-1} 1_{R_{<0}} \cdot (s_t \cdot s_{t-1}) \quad (1)$$

که S را در معادله بالا به صورت سیگنالی به طول T در نظر بگیریم و در ادامه $1_{R_{<0}}$ نیز تابعی برای مقادیر زیر صفر قابل برداشت خواهد بود.

در شکل نرخ گذر از صفر برای نمونه‌های مختلف از موسیقی‌های در کلاس‌های مختلف نشان داده شده است.



شکل ۴: نرخ گذر از صفر

جذر میانگین مربعات انرژی (RMSE): root-mean-square energy (RMSE)

انرژی در سیگنال‌های صوتی به صورت ساده شده برابر با اندازه‌ی سیگنال است. به عبارتی دیگر می توان گفت که انرژی تقریباً برابر به این است که چقدر یک صوت بلند است. جذر میانگین مربعات انرژی از رابطه زیر بدست می آید:

$$RMSE = \sqrt{\frac{1}{N} \cdot \sum_n |x(n)|} \quad (2)$$

آنتروپی شانون (Shannon entropy) :

آنتروپی کلاسیک شانون اطلاعات ارائه شده توسط مجموعه ای از رویدادها را اندازه گیری می کند و عدم قطعیت آن را نشان می دهد:

$$H(x) = - \sum_{i=1}^n p(x_i) \cdot \log_b p(x_i) \quad (2)$$

(۵-۲) ویژگی های حوزه فرکانس

اندازه طیف سیگنال:

نشان دهنده میزان انرژی در حوزه فرکانس است، به عبارت دیگر می توان از این مقدار متوجه شد که در هر فرکانس خاص، چه میزان انرژی وجود دارد. تبدیل فوریه زمان-کوتاه^۶ در این تبدیل سیگنال حوزه زمان را به پنجره های^۷ کوتاه مدت تقسیم می کند و از هر فریم تبدیل فوریه می گیرد.

به صورت کلی، بسیاری از سیگنال هایی که در کاربردهای عملی با آن ها مواجه هستیم به صورتی هستند که تغییرات آن ها در طول زمان را نمی توان به صورت دقیق توصیف کرد. در مورد این سیگنال ها فقط می توان عباراتی احتمالاتی برای توصیف تغییرات به کار برد. ابزار ریاضی برای مطالعه این سیگنال ها، همان ابزاری است که برای توصیف «دبیله های تصادفی» (Random Sequence) مورد استفاده قرار می گیرد که متشکل از گروهی از «تحقیق های» (Realizations) محتمل است که هر کدام دارای احتمال وقوع مختص به خود هستند. واضح است که از کل گروه تحقیق، هر «آزمایش کننده» (Experimenter) فقط یکی از تحقیق های سیگنال را مشاهده^۸ می کند. البته شایان ذکر است که ممکن است به این صورت تصور شود که می توان یک تعریف «قطعی» (Deterministic) را برای توصیف این سیگنال ها نیز مورد استفاده قرار داد. اما این تصور غلط است؛ زیرا تحقق یک سیگنال تصادفی به صورت یک دنباله گستته با زمان دیده می شود که انرژی محدود ندارد و به همین دلیل نمی توان از تبدیل فوریه گستته با زمان یا DTFT در مورد آن ها استفاده کرد. یک سیگنال تصادفی همواره دارای توان متوسط محدود است و بنابراین می توان با یک چگالی طیف توان میانگین آن ها را توصیف کرد. برای سادگی در ادامه به این مقدار فقط با نام چگالی طیف توان اشاره خواهیم کرد.

برای مشاهده بهتره اسپکتروگرام یک موسیقی در شکل زیر به همراه RMS رسم شده است.
STFT(Short-time Fourier transform):

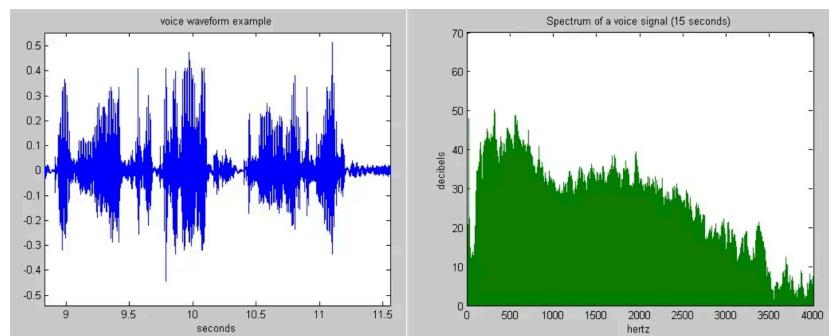
تبدیل فوریه کوتاه مدت، (STFT) یک تبدیل مرتبط با فوریه است که برای تعیین فرکانس سینوسی و محتوای فاز بخش های محلی سیگنال در طول زمان تغییر می کند. در عمل، روش محاسبه STFT ها به این صورت است که سیگنال زمان طولانی تری را به بخش های کوتاه تر با طول مساوی تقسیم می کند و سپس تبدیل فوریه را به طور جداگانه در هر بخش کوتاه تر محاسبه می کند. این طیف فوریه را در هر بخش کوتاه تر نشان می دهد.

سپس معمولاً یک طیف متغیر را به عنوان تابعی از زمان رسم می کند، که به عنوان طیف نگار یا نمودار آبشار شناخته می شود، مثلاً، معمولاً در نمایش های طیف میتنتی بر رادیو، نرم افزار (SDR) استفاده می شود. نمایشگرهای با پهنه ای باند کامل که کل محدوده یک SDR را پوشش می دهند.

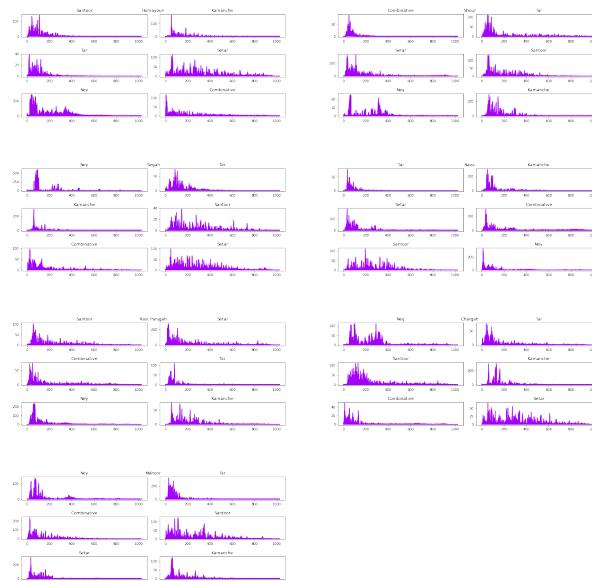
transform Fourier Short-time^۹

Frame^{۱۰}

Observe^{۱۱}



شکل ۵: اسپکتروگرام یک موسیقی



شکل ۶: نمونه ای از STFT

Continuous-time STFT

به سادگی، در حالت زمان پیوسته، تابعی که باید تبدیل شود در یک تابع پنجره که فقط برای مدت کوتاهی غیر صفر است ضرب می شود. تبدیل فوریه (یک تابع یک بعدی) سیگنال حاصل گرفته می شود، سپس پنجره در امتداد محور زمان لغزش می یابد تا انتها منجر به نمایش دو بعدی سیگنال شود. از نظر ریاضی به صورت زیر نوشته می شود:

$$STFTx(t)(\tau, \omega) = X(\tau, \omega) = \int_{-\infty}^{\infty} x(t) \cdot \omega(t - \tau) \cdot \exp^{-it\omega} \cdot dt \quad (4)$$

در معادله بالا ω تابع پنجره است و همچنین X سیگنال در حال تبدیل است و همان طور که مشخص است X تبدیل فوریه x است.

Discrete-time STFT

در مورد زمان گسسته، داده‌هایی که باید تبدیل شوند را می‌توان به تکه‌ها یا فریم‌هایی تقسیم کرد (که معمولاً روی یکدیگر همپوشانی دارند، تا مصنوعات در مز کاهش یابد). هر تکه تبدیل فوریه می‌شود و نتیجه مختلط به ماتریسی اضافه می‌شود که بزرگی و فاز را برای هر نقطه در زمان و فرکانس ثبت می‌کند. این را می‌توان به صورت زیر بیان کرد:

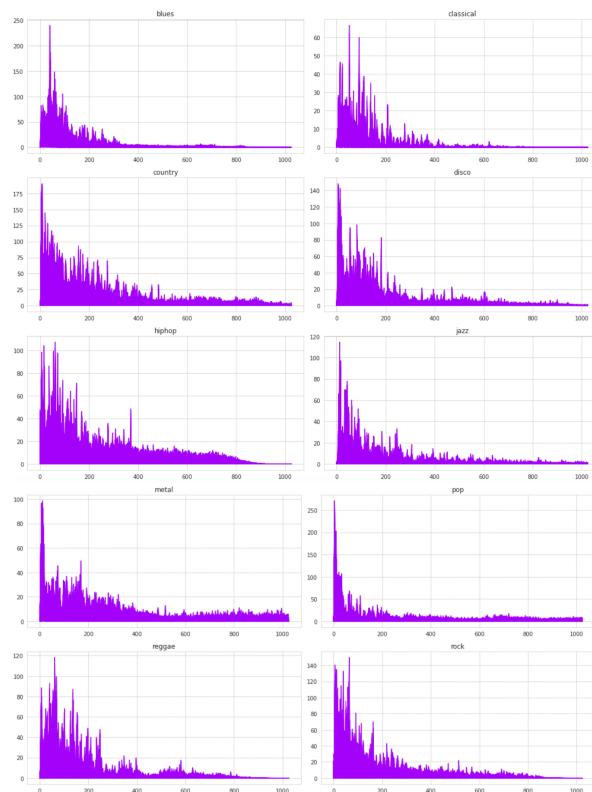
$$STFTx[n](m, \omega) = X(m, \omega) = \sum_{-\infty}^{\infty} x[n] \cdot \omega[n - m] \cdot \exp^{-in\omega} \quad (5)$$

که همانند گذشته تابع ω تابع پنجره و X سیگنال در حال تبدیل ولی این بار به صورت گسسته است.

حال با توجه به اطلاعاتی که به دست آورده‌یم به موضوع اوله خود بر می‌گردیم.

قدرتمندترین مقدار STFT طیف‌گرام چگالی طیفی توان تابع را به دست می‌دهد:

$$sepectrogram(x(t))(\tau, \omega) = |X(\tau, \omega)|^2 \quad (6)$$



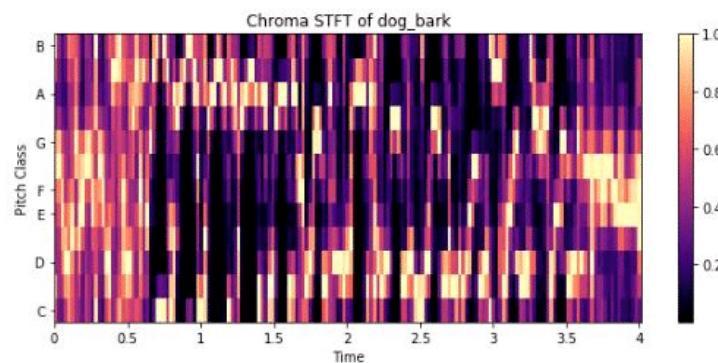
شکل ۷: نمودارهای فرکانس-شدت

Chroma SFTF

همان طوز که در شکل زیر نیز قابل مشاهده است، در این بخش از تبدیل فوریه کوتاه مدت که پیشتر به آن اشاره کردیم برای محاسبه

ویژگی های chroma استفاده می شود.

در ادمه STFT اطلاعات مربوط به طبقه بندی پیج و ساختار سیگنال را نشان می دهد. این نشان را با مقادیر پایین (همانطور که از شبکه نوار رنگ تا نمودار مشخص است) در مقادیر کم (مناطق تاریک) نشان داده می شود.



شکل ۸: ویژگی های کرومما

ویژگی های حوزه کپسٹرال:

یکی از ویژگی هایی که از سیگنال صوتی استخراج می شود و در بسیاری از کاربردها مورد استفاده قرار می گیرد ضرایب کپسٹرال می باشند.

آنالیز کپسٹرال Capstral Analysis روشی است متداول در حوزه پردازش سیگنال و پردازش گفتار صوتی به منظور فشرده سازی یا کد کردن گفتار با کیفیت بالا در نرخ بیت Bit rate پایین. این روش به طور گستردگی برای کد نمودن گفتار، سنتز گفتار، شناخت گفتار، شناخت گوینده، احراز هویت گوینده و ذخیره سازی گفتار استفاده می شود.

این ضرایب نه تنها اطلاعات فیلتر مجرای گفتار را در خود دارند، بلکه حاوی اطلاعات سیگنال تحریک نیز هستند و بنابراین مشخصه مناسبی برای گفتار، دسته بندی اصوات و ... می باشند.

از جمله مهمترین انواع آنالیز کپسٹرال می توان از موارد ذیل نام برد:

Complex cepstrum

Real cepstrum

Power cepstrum

Phase cepstrum

اساس کلی آنالیز کپسٹروم طی کردن روند زیر است: اما تفاوت انواع کپسٹروم در مرحله لگاریتم گیری اتفاق می افتد. به عنوان مثال برای



شکل ۹: مراحل آنالیز کپسٹروم

محاسبه Power ceptrum از لگاریتم اندازه به توان دو استفاده می کنیم، در محاسبه Real ceptrum از اندازه قسمت حقیقی DFT لگاریتم گرفته می شود و ...

تفاوت کپستروم حقیقی و مختلط این است که آنالیز کپستروم مختلط، حاوی اطلاعات فاز سیگنال صحبت نیز هست. اما از آنجا که اطلاعات فاز در شناوری انسان اهمیت اندکی دارد معمولاً از آنالیز کپستروم حقیقی استفاده می شود.

Harmonics and Perceptual

هارمونیک موج یا سیگنالی است که فرکانس آن ضرب انتگرال (عدد کامل) فرکانس همان سیگنال یا موج مرجع است. به عنوان بخشی از سری هارمونیک، این اصطلاح همچنین می تواند به نسبت فرکانس چنین سیگنال یا موجی به فرکانس سیگنال یا موج مرجع اشاره کند.

در مورد Perceptual می توان گفت که مربوط به احساسات و عاطفه می باشد که به صورت شهودی قابل بیان است.



شکل ۱۰: شمایی از Harmonics and Perceptual

Tempo BMP (beats per minute)

تعداد ضربات در یک دقیقه را نشان می دهد. به عنوان مثال، سرعتی که با ۶۰ BPM مشخص شده است به این معنی است که یک ضرب دقیقاً یک بار در ثانیه به صدا در می آید. سرعت ۱۲۰ ضربان در دقیقه دو بار سریعتر است، با دو ضربه در ثانیه.

Mel-Frequency Cepstral Coefficients

ضرایب مغزی فرکانس (MFCCs) یک سیگنال مجموعه کوچکی از ویژگی ها (ممکن است حدود ۲۰-۱۰) هستند که به طور خلاصه شکل کلی یک پوشش طیفی را توصیف می کنند. در MIR، اغلب برای توصیف تایم استفاده می شود.

Spectrogram

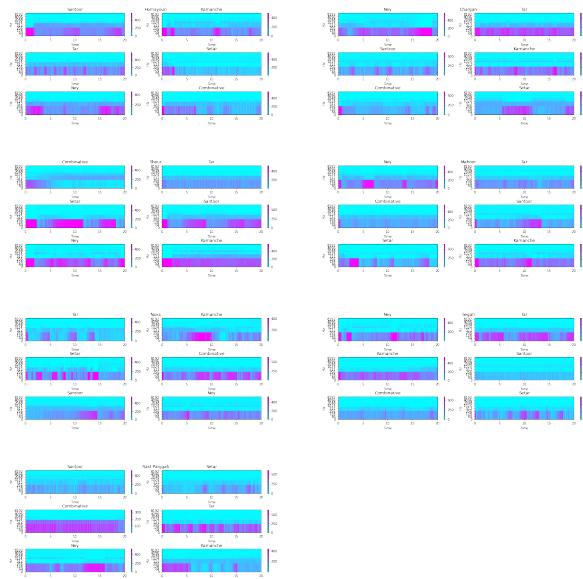
طیف‌نگار یک نمایش بصری از طیف فرکانس‌های یک سیگنال است که با زمان تغییر می‌کند. هنگامی که طیف‌نگارها بر روی یک سیگنال



شکل ۱۱: نمونه از نمایش تعداد ضربات در دقیقه

صوتی اعمال می‌شوند، گاهی اوقات سونوگرافی، چاپ صوتی یا صوت‌گرام نامیده می‌شوند. هنگامی که داده‌ها در یک نمودار سه بعدی نمایش داده می‌شوند، ممکن است نمایش آبشار نامیده شوند.
Mel Spectrogram :

یک طیف‌نگار mel به صورت لگاریتمی فرکانس‌ها را بالاتر از یک آستانه مشخص (فرکانس گوشه) ارائه می‌کند. به عنوان مثال، در طیف‌نگار با مقیاس خطی، فضای عمودی بین ۱۰۰۰ تا ۲۰۰۰ هرتز نیمی از فضای عمودی بین ۲۰۰۰ هرتز و ۴۰۰۰ هرتز است.



شکل ۱۲: نمایی از Mel-Frequency Cepstral

۶-۲) ویژگی های طیفی

برای طبقه بندی، ما از ویژگی های جدید در استفاده خواهیم کرد: گشتاورهای طیفی (مرکز، پهناهی باند، چولگی، کشیدگی) و سایر آمارهای طیفی.

در ادامه Momens اصطلاحی است که در فیزیک و آمار استفاده می شود. Momens محوری وجود دارد. احتمالاً قبلًا با دو مثال از Momens آشنا هستید: میانگین و واریانس. اولین Momens خام به عنوان میانگین شناخته می شود. دومین Momens مرکزی به عنوان واریانس شناخته می شود.

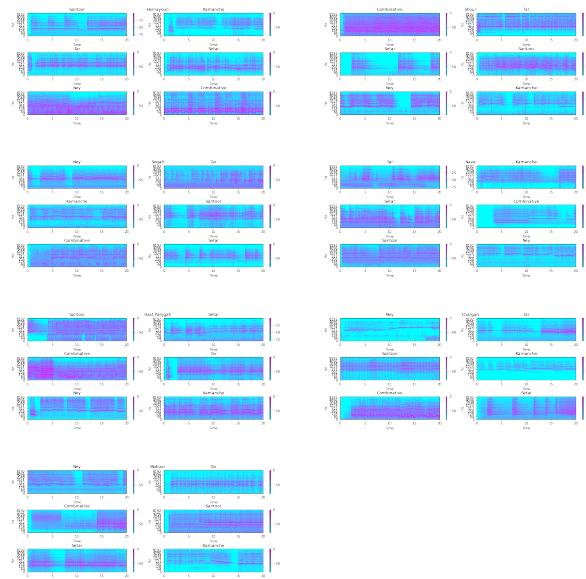
مرکز طیفی Spectral Centroid: مرکز طیفی نشان می دهد که انرژی یک طیف در کدام فرکانس متمرکز است. این مانند یک میانگین وزنی است:

$$f_c = \frac{\sum S(k) \cdot f(k))}{\sum S(k)} \quad (7)$$

در معادله بالا S قدر طیفی و f تابع فرکانس است.
پهناهی باند طیفی Bandwidth Spectral.

پهناهی باند طیفی در order p به صورت زیر محاسبه می شود:

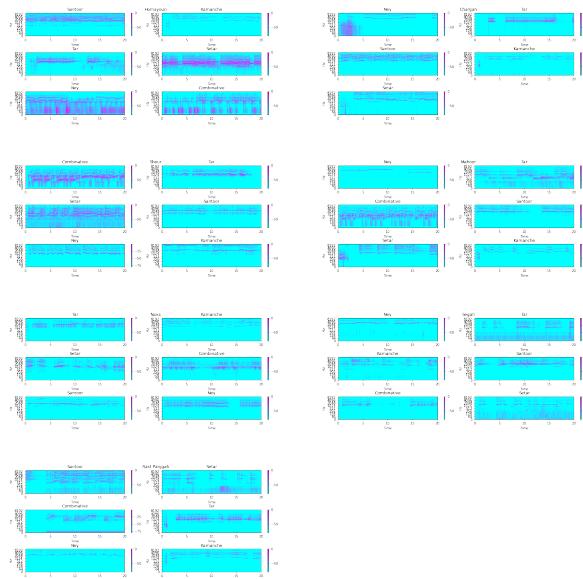
$$\left(\sum S(k) \cdot (f(k) - f_c)^p \right)^{\frac{1}{p}} \quad (8)$$



شکل ۱۳: نمایی از Spectrogram

که در معادله بالا S قدر طیفی و f تابع فرکانس است و f_c همان مرکز طیفی است.

کنتراست طیفی Spectral Contrast : کنتراست طیفی اوج طیفی، دره طیفی و تفاوت آنها را در هر زیر باند فرکانس در نظر می گیرد.



شکل ۱۴: نمایی از Mel Spectrogram

طیف رولوف Spectral Rolloff: طیفی رولوف فرکانسی را که در زیر آن درصد مشخصی از کل انرژی طیفی قرار دارد را مشخص می کند.

همچنین نمونه‌ی پیاده سازی ما از این طیف به صورت زیر به دست می آید:

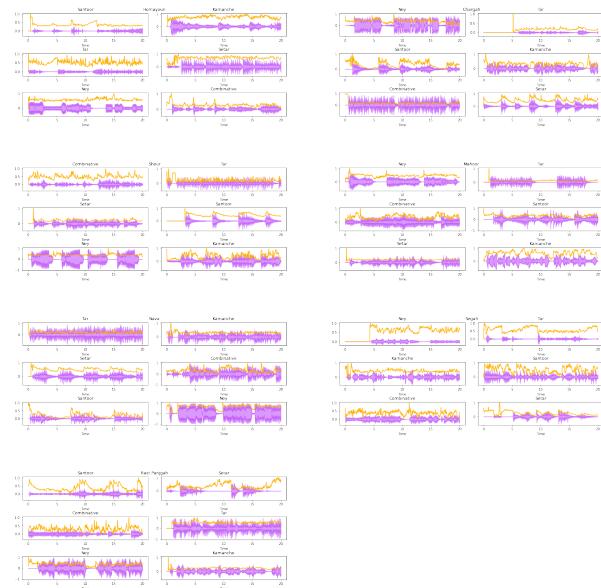
(۳) استخراج ویژگی و پیش پردازش داده ها

بعد از این که توانستیم آشنایی نسبی نسبت به این که یک سیگنال صوتی چیست، چه ویژگی هایی دارد و چگونه می توان به کمک این ویژگی ها آن ها را از هم تشخیص داد و طبقه بندی و خوش بندی کرد حال باید وارد وادی عملیاتی شده و شروع به پیاده سازی داشت خود بر روی داده ها نماییم.

(۱-۳) دیتاست

در این بخش داده های ما شامل ۶۴۸ عدد موسیقی بین ۲۰ ثانیه تا ۱۰ دقیقه است که ترکیبی از دستگاه ها و ساز های مختلف ایرانی است در هم ادامه اند و همچنین نوع ساز استفاده شده و دستگاه به کار رفته در موسیقی که درون دیتاست ما نهادینه شده است به صورت جدول زیر است.

همان طور که در شکل بالا مشاهده می شود دستگاه ها و ابزار های موسیقی نام برده شده شاکله کلی دیتاست مار تشکیل می دهند.



شکل ۱۵: نمایی از Spectral Centroid

در رابطه با این موضوع که هر دستگاه و هر ابزار موسیقی چه ویژگی هایی دارد و چگونه کار می کند به صورت کامل در گزارش اولیه لحاظ شده است و برای جلوگیری از زیاده گویی از تکرار آن صرف نظر می کنیم.

(۲-۳) استخراج ویژگی

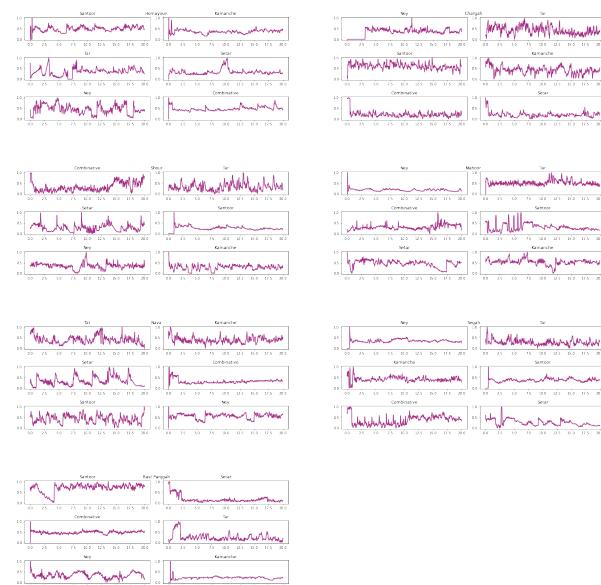
در ادامه با توجه به تمامی ویژگی های طیفی که در مورد سیگنال موسیقی بیان کردیم باید برای دیتاست خود که پیش تر اشاره کردیم شامل ۶۴۸ قطعه موسیقی که هر کدام بین ۲۰ ثانیه الی ۱۰ دقیقه هستند، به استخراج ویژگی از این دیتاست برسیم.

به طورکلی در پایتون از دو روش کلی برای استخراج ویژگی ها استفاده می کنیم که مفاهیم بخش دوم را به تصویر می کشند:
۱) librosa: یک بسته پایتون برای تجزیه و تحلیل موسیقی و صدا است. بلوک های ساختمانی لازم برای ایجاد سیستم های بازیابی اطلاعات موسیقی را فراهم می کند.

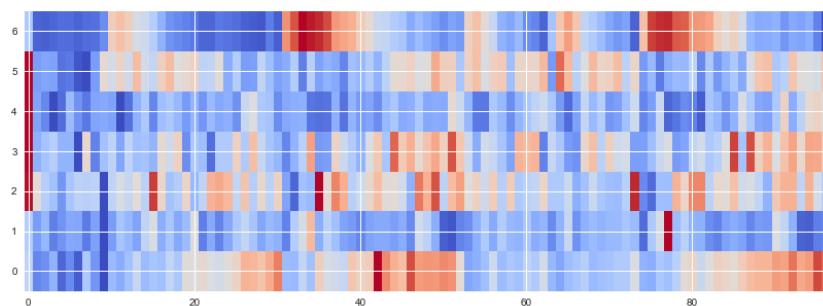
۲) text: یک کتابخانه پایتون برای استخراج ویژگی های است. این روش با هدف پرداختن به نقاط ضعف کتابخانه های موجود و تسهیل استفاده مشترک با چارچوب های مدرن یادگیری ماشین نوشته شده است.

حال بعد از بررسی کامل مراحلی که در بخش ۲ دیدم ویژگی ها مختلفی را از سیستم خود استخراج می کنیم که در ادامه مطلب با بررسی این ویژگی ها می توان از آنها در طبقه بندی و خوش بندی دیتاست خود استفاده کنیم که در این رابطع در ادامه به تفصیل صحبت خواهیم کرد.

ویژگی ها استخراج شده به شرح زیر خواهند بود:

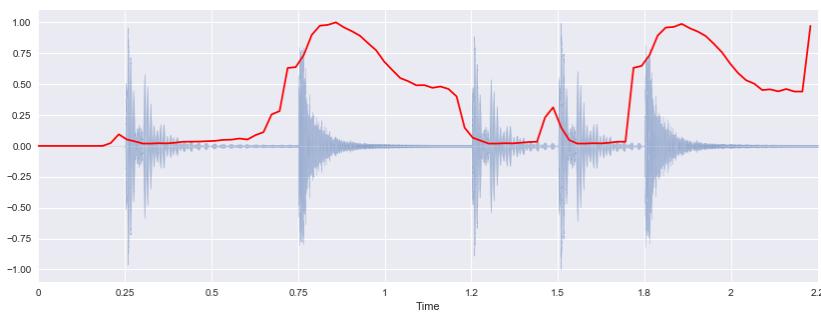


شکل ۱۶: نمایی از Spectral Bandwidth



شکل ۱۷: نمونه‌ای از کنترast طیفی

- | | | |
|-----------------------|-----------------------|------------------------|
| 1) chroma-stft-mean-1 | 2) chroma-stft-mean-3 | 3) chroma-stft-mean-5 |
| 4) chroma-stft-mean-7 | 5) chroma-stft-mean-9 | 6) chroma-stft-mean-11 |
| 7) chroma-stft-std-1 | 8) chroma-stft-std-3 | 9) chroma-stft-std-5 |
| 10) chroma-stft-std-7 | 11) chroma-stft-std-9 | 12) chroma-stft-std-11 |
| 13) rms-mean | 14) num-zerocrossing | 15) mfcc-mean-1 |
| 16) mfcc-mean-3 | 17) mfcc-mean-5 | 18) mfcc-mean-7 |
| 19) mfcc-mean-11 | 20) mfcc-mean-13 | 21) mfcc-std-2 |
| 22) mfcc-std-4 | 23) mfcc-std-6 | 24) mfcc-std-8 |
| 25) mfcc-mean-10 | 26) mfcc-mean-12 | 27) mfcc-skewness-1 |
| 28) mfcc-skewness-3 | 29) mfcc-skewness-5 | 30) mfcc-skewness-7 |



شکل ۱۸: نمونه‌ای از طیف رولوف

- | | | |
|---------------------|----------------------|----------------------|
| 31) mfcc-skewness-9 | 32) mfcc-skewness-11 | 33) mfcc-skewness-13 |
| 34) mfcc-kurtosis-2 | 35) mfcc-kurtosis-4 | 36) mfcc-kurtosis-6 |
| 37) mfcc-kurtosis-8 | 39) mfcc-kurtosis-10 | 40) mfcc-kurtosis-12 |

همان طور که در بالا قابل مشاهده است ۴۰ ویژگی از دیتاست خود با تمامی اطلاعاتی که در بخش قبل به آن اشاره کردیم استخراج می‌کنیم که شکل و چهار چوب آن از نام آن‌ها مشخص است.

در ادامه باید به این مسیله پردازیم که تاثیر این ویژگی‌ها و ارتباط آن‌ها با دستگاه‌ها را مورد بررسی قرار می‌دهیم به نحوی که ببینیم کدام ویژگی می‌تواند در ادامه خواسته‌های ما را بهتر برآورد کند و کدام ویژگی‌ها در برآورده کردن خواست ما توان کمتری دارد. برای فهمیدن این موضوع از correlation matrix مربوط به این ویژگی‌ها استفاده می‌کنیم.

همان طور هم که می‌دانیم correlation matrix یا ماتریس همبستگی جدولی است که ضرایب همبستگی را برای متغیرهای مختلف نشان می‌دهد. ماتریس همبستگی بین تمام جفت مقادیر ممکن در یک جدول را نشان می‌دهد. این ابزار قدرتمندی برای خلاصه کردن یک مجموعه داده بزرگ و شناسایی و تجسم الگوهای در داده‌های داده شده است. با توجه به آن چه که گفته شد ماتریس همبستگی ما به صورت زیر به دست می‌آید:



شکل ۱۹: نمایی از Spectral Roll-off

همان طورکه از ماتریس همبستگی مشخص است بین ویژگی ها و دستگاه correlation زیادی دیده نمی شود پس می توان فهمید برای طبقه بندی داده ها به چیزی بیشتر از یک مدل خطی نیاز داریم که در ادake بیشتر به آن می پردازیم. در ادامه نکته مهم دیگری که باید به آن پردازیم نحوه توزیع داده است زیرا اگر نحوه توزیع داده ها خیلی از هم متفاوت باشد باید از طریق وزن دهنده عملیات را پیش ببریم. با توجه به این موضوع نحوه توزیع داده ها برای دستگاه ها و ابزار های موسیقی مختلف که بیشتر در رابطه با آن ها صحبت کردیم به صورت زیر به دست می آید:

Instrument	Instrument ID	Dastgah	Dastgah ID
تار	۰	شور	۰
کمانچه	۱	سیگاه	۱
ستور	۲	ماهور	۲
ستمار	۳	همایون	۳
نی	۴	راست پنجگاه	۴
ترکیبی	۵	ثوا	۵
		چهارگاه	۶

شکل ۲۰: ویژگی های دیتاست

همان طور که مشخص است از آن جهت که نحوه تقسیم داده ها به نحوی است که خیلی اختلاف ندارند نیاز نیست ضریب خطایی به سیستم بدھیم و از این قسمت به بعد می توان وارد بخش ایجاد مدل برای طبقه بندی داده ها شویم.

(۳-۳) پیش پردازش داده ها

در این بخش پیش این که وارد بخش مدل سازی و طبقه بندی داده های ما نیازمند پردازش اولیه هستند تا بتوان عملیات مدل سازی را بر روی آن ها انجام داد. به عبارت دیگر، پیش پردازش داده فرآیندی است برای تهیه داده های خام و مناسب ساختن آن برای یک مدل یادگیری ماشینی. این اولین و مهم ترین گام در ایجاد یک مدل یادگیری ماشینی است. هنگام ایجاد یک پروژه یادگیری ماشینی، همیشه با داده های تمیز و فرمت شده مواجه نمی شویم.

مراحل پیش پردازش داده ها: حال در ادامه به مراحل مشخصی بیندازیم که باید طی کنید تا مطمئن شوید داده های شما با موفقیت پیش پردازش شده است.

Data quality assessment (۱)

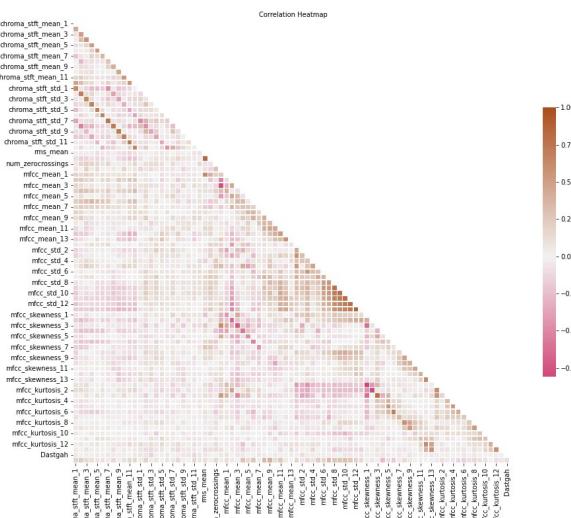
Data cleaning (۲)

Data transformation (۳)

Data reduction (۴)

: داده ها را مورد بررسی قرار می دهیم و از کیفیت کلی، ارتباط با پروژه خود و سازگاری آن ایده می گیریم. تعدادی از ناهنجاری های داده و مشکلات ذاتی وجود دارد که تقریباً در هر مجموعه داده ای باید به دنبال آنها باشیم..

: پاکسازی داده ها ، فرآیند افزودن داده های از دست رفته و تصحیح، تعمیر یا حذف داده های نادرست یا نامربوط از یک مجموعه داده است. پاکسازی داده ها مهمترین مرحله پیش پردازش است زیرا اطمینان حاصل می کند که داده های ما برای نیازهای

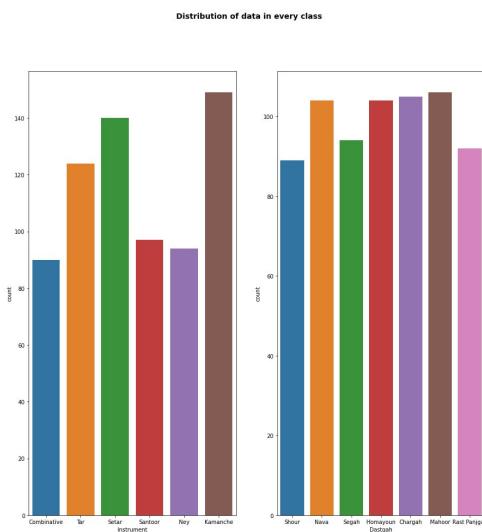


شکل ۲۱: ماتریس همبستگی

پایین آمده است.

با پاکسازی داده‌ها، ما قبل شروع به اصلاح داده‌های خود کردیم، اما فرآیند تبدیل داده‌ها به قالب (های) مناسبی که برای تجزیه و تحلیل و سایر فرآیندهای پایین دستی نیاز داریم، آغاز می‌کنیم.

Data reduction : هرچه باداده‌های بیشتری کار کنیم، حتی پس از تمیز کردن و تبدیل آن‌ها، تجزیه و تحلیل آن سخت‌تر خواهد بود. بسته به وظیفه ای که در دست داریم، ممکن است در واقع داده‌های بیشتری از آنچه نیاز داریم داشته باشیم. به خصوص هنگام کار با تجزیه و تحلیل متن، بسیاری از گفتار منظم انسان اضافی یا بی ربط به نیازهای محقق است. کاهش داده‌ها نه تنها تجزیه و تحلیل را آسان‌تر و دقیق‌تر می‌کند، بلکه ذخیره سازی داده‌ها را کاهش می‌دهد.



شکل ۲۲: توزیع داده ها

در اینجا ما یک استانداردسازی جایگزین و مقیاس‌بندی ویژگی‌ها برای قرار گرفتن بین حداقل و حدکثر مقدار معین، اغلب بین صفر و یک، قرار می‌دهیم که به گونه‌ای است که حدکثر مقدار مطلق هر ویژگی به اندازه واحد مقیاس شود. انگیزه استفاده از این مقیاس‌بندی شامل استحکام تا انحرافات استاندارد بسیار کوچک ویژگی‌ها و حفظ صفر و رودی در داده‌های پراکنده است.

(۴-۳) کاهش بعد

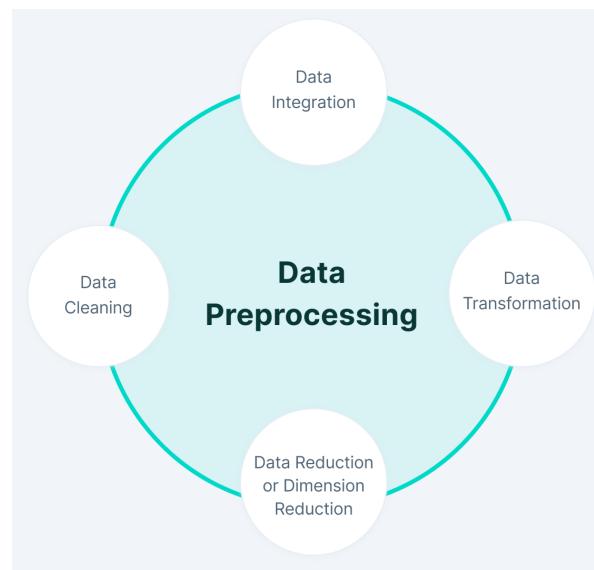
پس از استخراج ویژگی‌ها، از آنجایی که تعداد زیادی ویژگی برای هر صوت داریم، با مشکل پیچیده شدن مدل و سخت شدن طبقه‌بندی مواجه هستیم. برای حل این مشکل روش‌هایی همچون کاهش بعد برای کم کردن تعداد ویژگی‌هایی داریم. در این پروژه از روش‌های زیر برای کاهش ابعاد داده‌ها استفاده کردایم:

PCA •

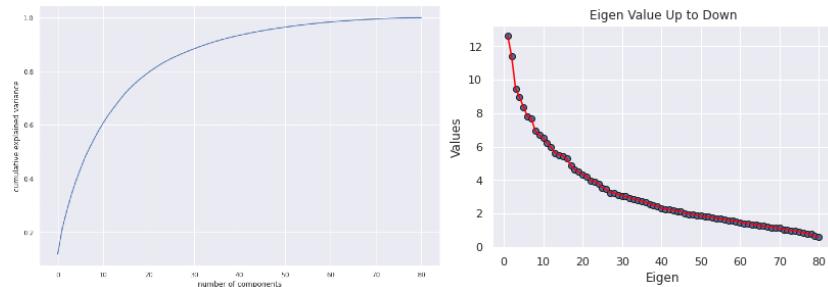
روشی بدون نظرارت است که قصد دارد با کمتر کردن تعداد ابعاد داده، کار طبقه‌بندی را ساده تر کند. به عبارت دیگر این روش داده‌ها را به ابعادی دیگر تصویر می‌کند که واریانس آن بعد بیشترین حد باشد. با توجه به دو نمودار زیر می‌توان ایده‌ای از تعداد های PCA مورد نیاز برای مجموعه داده خودم بگیریم. با توجه به این شکل متوجه می‌شویم که ۳۰ عدد از ویژگی‌های انتخاب شده ۹۰ درصد واریانس داده‌ها را در بر می‌گیرند. همچنین با توجه به مقدار ویژه‌های بدست آمده مشخص است که شبیه اطلاعات از حدود ۳۰ ویژگی به بعد تقریباً کم شده و در نتیجه ویژگی‌ها ممکن است اطلاعات زیادی برای طبقه‌بندی فراهم نکنند.

LDA •

این روش با نظرارت است که با توجه به کلاس داده‌ها سعی می‌کند ویژگی‌هایی را انتخاب کند که بیشترین جدایی‌پذیری بین کلاسی و بیشترین شباهت درون کلاسی را ایجاد کند. با توجه به نمودار زیر می‌توان ایده‌ای از تعداد های LDA مورد نیاز برای مجموعه



شکل ۲۳: پیش پردازش داده



شکل ۲۴: PCA

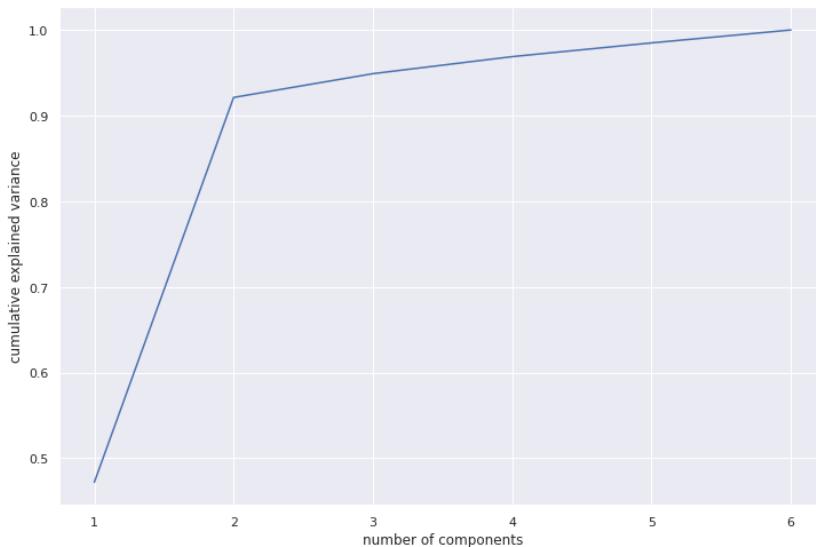
داده خود بگیریم. با توجه به این شکل متوجه می‌شویم که حدود ۵ تا از ویژگی‌ها ۹۰ درصد واریانس داده‌ها را در بر می‌گیرند.

Sequential feature selection •

روش Sequential (SFS) selection forward قصد دارد زیرمجموعه‌ای از ویژگی‌ها پیدا کند. این روش به صورت حریصانه در هر مرحله بهترین ویژگی را انتخاب می‌کند. به عبارتی در هر مرحله با استفاده از تخمین زنی (مدل) تک تک همه ویژگی‌ها را امتحان کرده و ویژگی که بهترین عملکرد را می‌دهد انتخاب می‌کنیم و به زیرمجموعه‌ی خود اضافه می‌کنیم و اینکار را تا جای لازم انجام می‌دهیم. نمودار زیر اثر تعداد ویژگی‌ها و افزایش آن‌ها در خطأ و دقت مدل را نشان می‌دهد.

حال در قسمت کدنویسی بعد از پیاده سازی این روش برای کاهش ویژگی‌ها به ۱۰ عدد نتیجه به دست آمده به صورت زیر

خواهد بود: (۱) chroma-stft-mean-1 (۲) chroma-stft-mean-2 (۳) chroma-stft-mean-5
 (۴) chroma-stft-mean-6 (۵) chroma-stft-std-3 (۶) chroma-stft-std-4 (۷) chroma-stft-std-8 (۸) chroma-stft-std-9 (۹) chroma-stft-std-11 (۱۰) chroma-stft-mean-6



شکل ۲۵: LDA

همان طور که مشاهده می شود بعد کاهش ویژگی ها به روش Feature Selection ویژگی های انتخابی به صورت بالا خواهد بود.

حال در قسمت کدنویسی بعد از پیاده سازی این روش برای کاهش ویژگی ها به ۱۰ عدد نتیجه به دست آمده به صورت زیر خواهد بود:

chroma-stft-mean-5 (۳)	chroma-stft-mean-2 (۲)	chroma-stft-mean-1 (۱)
chroma-stft-std-۴ (۶)	chroma-stft-std-3 (۵)	chroma-stft-mean-8 (۴)
chroma-stft-std-11 (۹)	chroma-stft-std-9 (۸)	chroma-stft-mean-6 (۷)

همان طور که مشاهده می شود بعد کاهش ویژگی ها به روش Feature Selection ویژگی های انتخابی به صورت بالا خواهد بود.

Backward feature elimination •

روش BFE قصد دارد زیرمجموعه ای از ویژگی ها پیدا کند. این روش ابتدا تخمین گری با استفاده از تمام ویژگی ها آموخت داده و در هر مرحله به صورت حریصانه در هر مرحله ویژگی که کمترین کاربرد را دارد حذف می کند. نمودار زیر اثر تعداد ویژگی ها و کاهش آن ها در خط اطلاعاتی دقت مدل را نشان می دهد.

ه از این روش یا به عبارتی Backward Elimination با هدف رسیدن به ۱۰ ویژگی به نتایج زیر دست پیدا می کنیم:

chroma-stft-std-3 (۳)	chroma-stft-std-2 (۲)	chroma-stft-mean-3 (۱)
chroma-stft-std-9 (۶)	chroma-stft-std-7 (۵)	chroma-stft-std-4 (۴)
mfcc-mean-2 (۹)	chroma-stft-std-12 (۸)	chroma-stft-std-11 (۷)

mfcc-mean-7 (۱۰)

همان طورکه مشاهده می شود با توجه به روش Backward Elimination به نتایج متفاوتی می رسیم که در بالا مشاهده شود.

(۵-۳) انتخاب ویژگی

انتخاب ویژگی روشنی برای کاهش متغیر ورودی به مدل شما با استفاده از داده های مربوطه و خلاص شدن از شر نویز در داده ها است. این فرآیند انتخاب خودکار ویژگی های مرتبط برای مدل یادگیری ماشین شما بر اساس نوع مشکلی است که می خواهید حل کنید.

(۴) طبقه بندی کننده ها

طبقه بندها مدل های یادگیری ماشین نظارت شده ای هستند که بر اساس داده ها و کلاس هایشان آموزش می بینند. در این بخش چند طبقه بند از جمله شبکه عصبی چند لایه، Logistic Regression و SVM آموزش داده شده اند. و نتایج در هر قسم مقایسه و توضیح داده شده است. شایان ذکر است که قبل از شروع عملیات طبقه بندی مجموعه داده خود را به دو بخش آموزش^۹ و تست^{۱۰} تقسیم می کنیم تا بتوانیم با آموزش دادن مدل خود بر روی داده های آموزش و بررسی نتایج بر روی داده های تست، تعیین پذیری مدل ها را نیز بررسی می کنیم.

(۱-۴) شبکه عصبی چند لایه

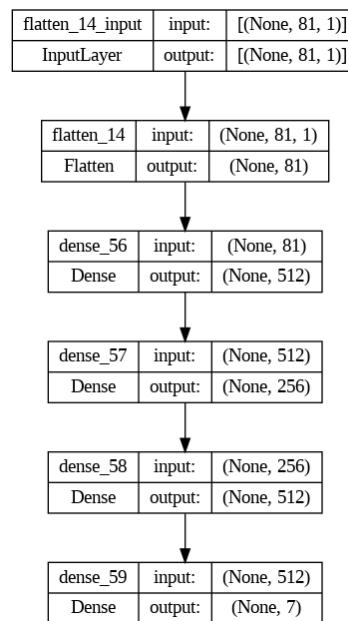
در این قسمت شبکه عصبی به منظور طبقه بندی این داده ها آموزش می دهیم. معماری این شبکه عصبی توسط [?] پیشنهاد شده است. این شبکه عصبی با استفاده از کتابخانه Tensorflow پیاده سازی شده است. لازم یه ذکر است که علاوه بر داده های آموزش و تست در این بخش نیز ۱۰ درصد از داده های آموزش برای ارزیابی^{۱۱} مشخص کردن های پارامترها در نظر گرفته شده است. در طی روند آموزش با استفاده از Callback در مدلی که بهترین نتیجه را روی داده ارزیابی دارد نگه می داریم. این شبکه عصبی دارای ۶ لایه می باشد که لایه اول ۵۱۲ نورون دارد و لایه آخر ۷ نورون (به تعداد کلاس ها) همچنین لایه dropout به منظور جلوگیری از overfit شدن استفاده شده است. توابع فعال ساز در لایه های اول ReLU و در لایه آخر Softmax در نظر گرفته شده است. همچنین با توجه به اینکه در حال طبقه بندی چند کلاسه هستیم entropy cross انجام می دهیم. علاوه بر این تابع بهینه ساز برای خطای نیز adams با نرخ یادگیری ۰.۰۱ در نظر گرفته ایم. معماری شبکه به شکل زیر می باشد.

و همچنین ساختار لایه ای آن به صورت زیر قابل مشاهده است:

Train [*]	
Test ^{۱۱}	
Validation ^{۱۱}	

Params	Shape	Layer
.	۸۱	Flatten
۴۱۹۸۴	۵۱۲	Dense
۱۳۱۳۲۸	۲۵۶	Dense
۱۳۱۵۸۴	۵۱۲	Dense
۳۵۹۱	۷	Dense

جدول ۱: مدل MLP



شکل ۲۶: مدل MLP

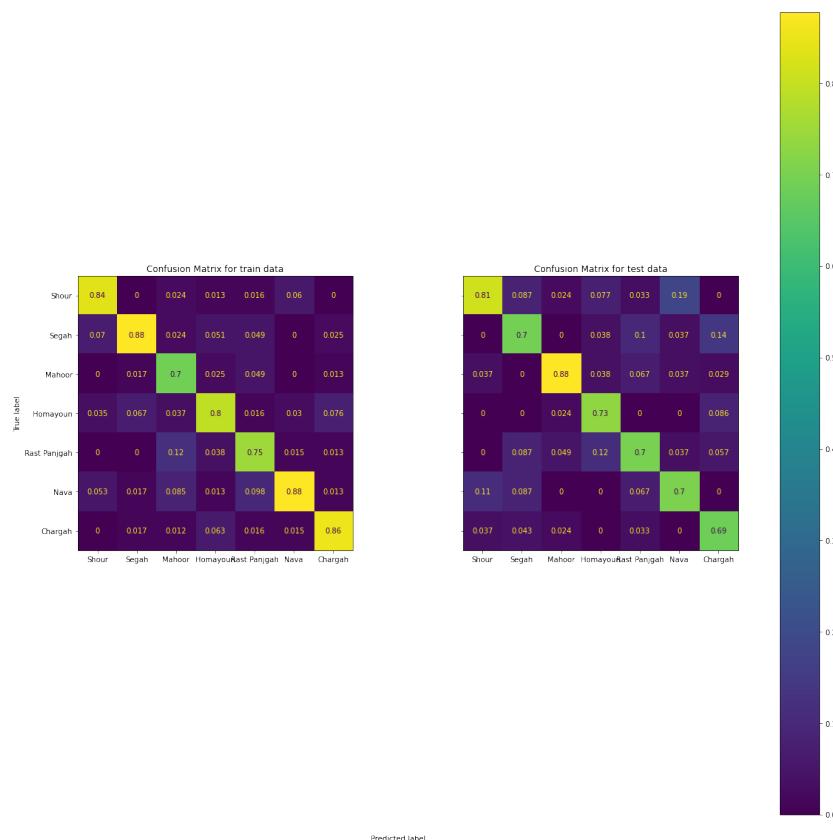
بعد از پیاده سازی این مدل ترتیب به دست آمده در رابطه با این مدل MLP و همچنین با طی کردن ۱۰۰ تعدد دفعات صورت زیر خواهد بود:

مدل شبکه عصبی چندلایه، نسبتاً به دقت خوبی دست می‌یابد. این موضوع با توجه به ماهیت شبکه‌های عصبی قابل پیش‌بینی بود. این مدل‌ها با تصویر کردن داده‌ها به بعدهای مختلف می‌توانند بهترین ناحیه جدایزیری برای داده‌ها را پیدا کنند و این وظیفه را بهترین حالت انجام دهنند. البته اگر به ماهیت داده‌ها توجه کنیم، برای سیگنال‌های صوتی بهتر از شبکه‌های عصبی حافظه‌دار همچون LSTM استفاده کنیم. نمودار آموزش شبکه‌ی عصبی به شکل زیر می‌باشد: حال در ادامه ماتریس پراکندگی مربوط به داده‌های تست و آموزش رارسم می‌کنیم کلی از عملکرد سیستم بر روی داده‌ها را به نمایش بگذارد.

ماتریس پراکندگی بر روی داده‌های آموزش و تست به صورت زیر به دست می‌آید:

Loss	Accuracy	Model
0.08047939091920853	0.9904305934906006	MLP

جدول ۲: نتایج MLP



شکل ۲۷: ماتریس پراکندگی MLP

ماتریس پراکندگی بهتر نحوه تقسیم‌بندی کلاس‌ها را نشان می‌دهد. برای مثال در بعضی از کلاس‌ها که تا حدی شبیه هم هستند شبکه اشتباهات بیشتری نسبت به سایر کلاس‌ها انجام می‌دهد.

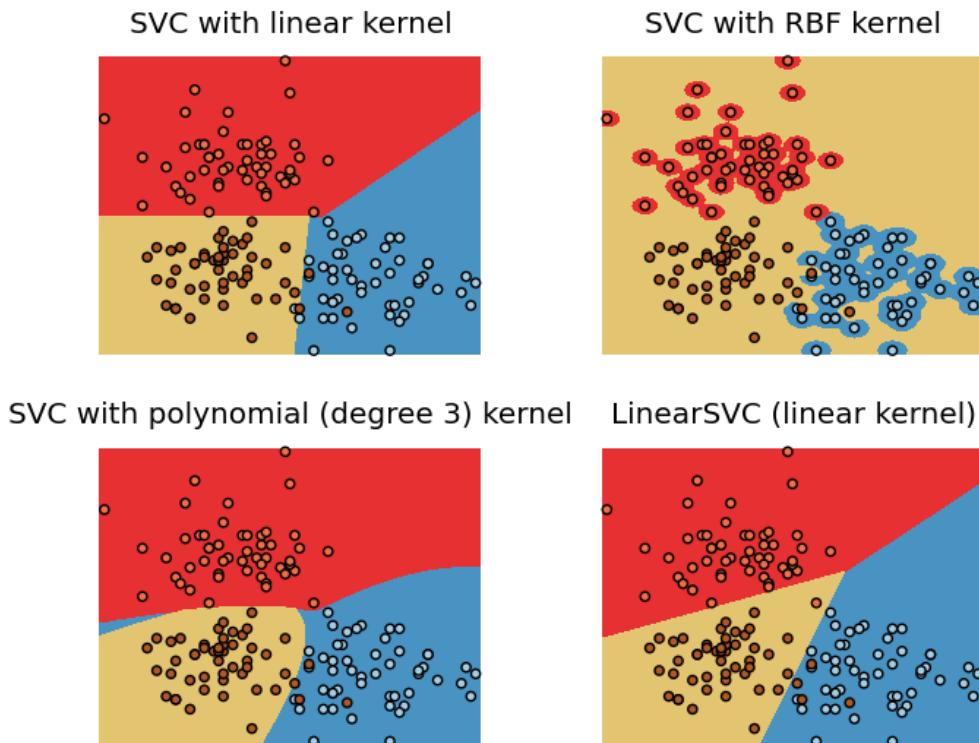
SVM (۲-۴)

به طورکلی Support vector machines(SVM) مجموعه‌ای از روش‌های یادگیری تحت نظارت (supervised) هستند که برای طبقه‌بندی، رگرسیون استفاده می‌شوند. همه اینها وظایف رایج در یادگیری ماشین هستند. انواع خاصی از SVM ها وجود دارد که می‌توانید برای مشکلات خاص یادگیری ماشین استفاده کنید، مانند support vector classification (SVC)، support vector regression (SVR) که توسعه (SVM) است.

در SVM ها با سایر الگوریتم‌های طبقه‌بندی متفاوت هستند، زیرا آنها مرز تصمیم را انتخاب می‌کنند که فاصله از نزدیک‌ترین نقاط داده

همه کلاس‌ها را به حداقل می‌رساند.

نحوه عملکرد کلی SVM به این صورت است که با استفاده از کرnel‌های داده‌ها را به بعدهای دیگری منتقل می‌کند و در این نواحی جدید تقسیم‌بندی را انجام می‌دهد. البته باید توجه داشت که با استفاده از کرnel‌های مختلف نتایج ما فرق خواهد کرد، برای مثال با توجه به نحوه توزیع داده‌ها مشخص است که برای داده‌های ما به صورت خطی قابل جداسازی نخواهد بود لذا بهتر است از کرnel‌هایی مانند Poly و RBF استفاده کنیم.



شکل ۲۸: کرnel های مختلف SVM

پیش از آن که در مورد مدل پیاده سازی شده صحبت کنیم باید اشنایی نسبی در مورد دو روش One-VS-All و One-VS-One داشته باشیم که به صورت خلاصه در مورد آن صحبت می‌کنیم.

One-VS-All (OvA): می‌توانیم One-Vs-Rest (OvR) One-Vs-Rest را به عنوان رویکردی برای ساختن الگوریتم‌های طبقه‌بندی باینری که قادر به کار به عنوان الگوریتم‌های طبقه‌بندی چند کلاسه هستند، در نظر بگیریم. این رویکرد عمدهاً داده‌های چند کلاسه را به عنوان داده‌های طبقه‌بندی باینری تقسیم می‌کند تا الگوریتم‌های طبقه‌بندی باینری را بتوان برای تبدیل داده‌های طبقه‌بندی باینری اعمال کرد.

One-VS-One(۲): این روش همچنین می تواند به عنوان رویکردی برای ساخت الگوریتم های طبقه بندی بایزی با قابلیت کار به عنوان الگوریتم های طبقه بندی چند کلاسه در نظر گرفته شود. این روش مشابه روش One-Vs-Rest است زیرا بر اساس تقسیم داده ها نیز کار می کند، اما رفتار تقسیم این روش به این صورت است. متفاوت از روش One-Vs-Rest. این روش شامل تقسیم داده ها برای هر کلاس است که در آن هر کلاس هر کلاس دیگری را به عنوان حریف خود دارد. ما دوباره می توانیم از مجموعه داده های عنبیه برای درک رفتار تقسیم داده های این روش استفاده کنیم.

در این بخش شکل های مختلفی از SVM را پیده سازی می کنیم و نتایج مربوط به هر کدام را به صورت جداگانه مورد بررسی قرار می دهیم:

Linear SVM One-Vs-Rest(۱)

Linear SVM One-Vs-One(۲)

rbf SVM (gamma=0.7)(۳)

poly SVM (dgree=3)(۴)

حال نتایج هر کدام از این مدل ها را به صورت جداگانه بر روی دیتابیست خود یا به عبارتی بر روی داده های train و test به صورت جداگانه مورد بررسی قرار می دهیم مورد بررسی قرار می دهیم که به صورت زیر تعریف می شوند.

نتایج train بر روی دیتابی ای Linear SVM One-Vs-Rest

support	f1-score	recall	precision	
69	0.88	0.83	0.95	شور
61	0.98	0.97	1.00	سه گاه
83	0.91	0.88	0.94	ماهور
81	0.84	0.84	0.85	همایون
56	0.9	0.96	0.84	راست پنج گاه
67	0.84	0.88	0.81	نوا
68	0.92	0.94	0.9	چهار گاه
				Accuracy
485	0.89			macro avg
485	0.9	0.9	0.9	weighted avg

جدول ۳: نتایج train بر روی دیتابی ای Linear SVM One-Vs-Rest

نتایج test بر روی دیتابی ای Linear SVM One-Vs-Rest

support	f1-score	recall	precision	
41	0.34	0.29	0.41	شور
26	0.30	0.35	0.26	سه گاه
40	0.26	0.23	0.32	ماهور
27	0.16	0.15	0.17	همایون
21	0.16	0.19	0.14	راست پنج گاه
30	0.23	0.23	0.23	نوا
24	0.41	0.5	0.35	چهار گاه
				Accuracy
209	0.27			macro avg
209	0.27	0.27	0.28	weighted avg

جدول ۴: نتایج Linear SVM One-Vs-Rest بر روی دیتای test

نتایج train بر روی دیتای Linear SVM One-Vs-One

support	f1-score	recall	precision	
55	0.49	0.51	0.47	شور
73	0.73	0.66	0.81	سه گاه
81	0.68	0.67	0.69	ماهور
70	0.59	0.63	0.55	همایون
58	0.67	0.60	0.55	راست پنج گاه
71	0.61	0.62	0.60	نوا
77	0.70	0.68	0.73	چهار گاه
				Accuracy
485	0.63			macro avg
485	0.63	0.63	0.64	weighted avg

جدول ۵: نتایج train بر روی دیتای Linear SVM One-Vs-One

نتایج test بر روی دیتای Linear SVM One-Vs-One

support	f1-score	recall	precision	
26	0.25	0.27	0.24	شور
33	0.32	0.33	0.31	سه گاه
38	0.24	0.21	0.29	ماهور
24	0.17	0.17	0.17	همایون
26	0.19	0.19	0.18	راست پنج گاه
31	0.26	0.26	0.26	نوا
31	0.40	0.42	0.38	چهار گاه
209	0.27			Accuracy
209	0.26	0.26	0.26	macro avg
209	0.27	0.27	0.27	weighted avg

جدول ۶: نتایج Linear SVM One-Vs-One بر روی دیتای test

نتایج train بر روی دیتای rbf SVM

support	f1-score	recall	precision	
65	1	1	1	شور
59	1	1	1	سه گاه
78	1	1	1	ماهور
80	1	1	1	همایون
64	1	1	1	راست پنج گاه
73	1	1	1	نوا
71	1	1	1	چهار گاه
485	1			Accuracy
485	1	1	1	macro avg
485	1	1	1	weighted avg

جدول ۷: نتایج train rbf SVM بر روی دیتای

نتایج test بر روی دیتای rbf SVM

support	f1-score	recall	precision	
1	0.07	1	0.03	شور
1	0.06	1	0.03	سه گاه
2	0.07	0.50	0.04	ماهور
203	0.20	0.11	0.96	همایون
2	0.13	1	0.07	راست پنج گاه
0	0.00	0.00	0.00	نوا
0	0.00	0.00	0.00	چهار گاه
				Accuracy
209	0.13			macro avg
209	0.20	0.13	0.93	weighted avg

جدول ۸: نتایج test rbf SVM بر روی دیتای

train نتایج poly SVM بر روی دیتای

support	f1-score	recall	precision	
35	0.74	1	0.58	شور
33	0.72	1	0.56	سه گاه
80	0.94	0.93	0.95	ماهور
78	0.94	0.95	0.93	همایون
48	0.86	1	0.75	راست پنج گاه
155	0.64	0.47	1	نوا
56	0.88	1	0.79	چهار گاه
				Accuracy
485	0.81			macro avg
485	0.80	0.81	0.87	weighted avg

جدول ۹: نتایج train poly SVM بر روی دیتای

test نتایج poly SVM بر روی دیتای

support	f1-score	recall	precision	
7	0.28	0.71	0.17	شور
3	0.05	0.33	0.03	سه گاه
40	0.35	0.30	0.43	ماهور
20	0.14	0.15	0.12	همایون
9	0.22	0.44	0.14	راست پنج گاه
113	0.35	0.22	0.81	نوا
17	0.47	0.71	0.35	چهار گاه
209	0.30			Accuracy
209	0.26	0.41	0.29	macro avg
209	0.33	0.30	0.57	weighted avg

جدول ۱۰: نتایج poly SVM بر روی دیتای test

همان طور که مشاهده می شود نتایج برای تقریبا همه ها train SVM قابل قبول است ولی به همان ترتیب بر روی داده های تست نتایج بهترین حالت نیست. اگر بخواهیم نتایج هر یک از چهار مدل را به صورت جداگانه بر روی داده های آموزش مورد بررسی قرار دهیم باید اشاره کرد، همان طور که از قبل پیش‌بینی نیز می شد بدترین نتیجه را SVM خطی OvO برای ما به دست می آورد زیرا توانای لازم برای طبقه بندی داده های ما با توجه به ویژگی هایی که پیشتر لحاظ کردیم را ندارد و در محاسبات خود برای طبقه بندی از ساختار ساده تری استفاده می کند در ادامه البته باید اشاره کرد که SVM خطی OvR بر روی داده های آموزش نتیجه به مراتب بهتری به ما ارایه می دهد.

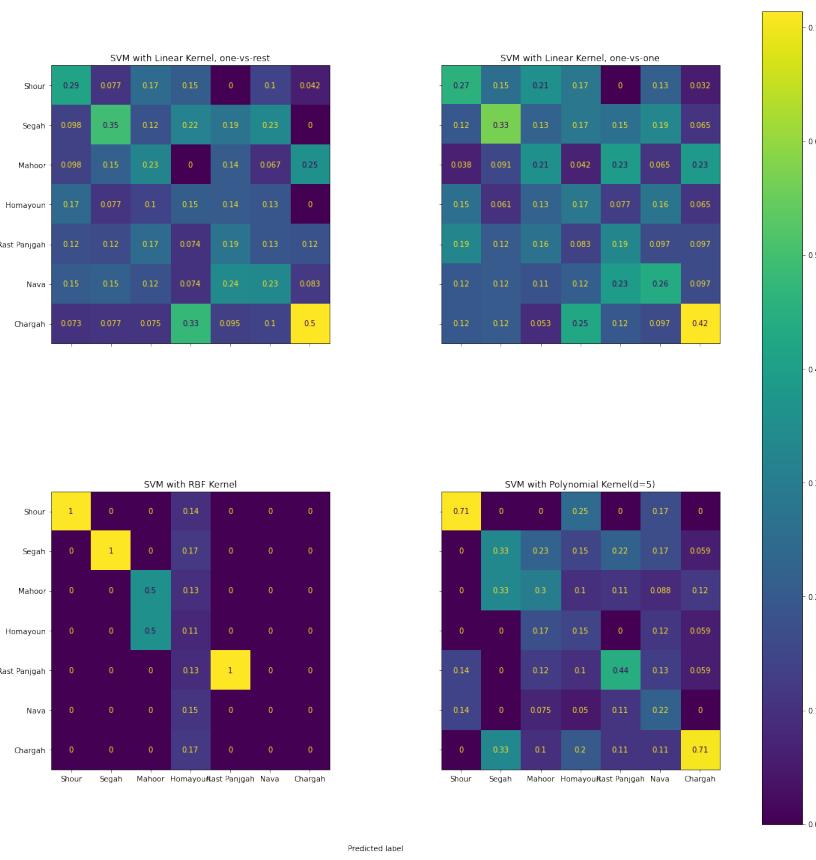
در ادامه شایان ذکر است که poly SVM بر روی داده های train عملکردی مشابه SVM خطی OvR دارد و تفاوت چندانی با هم پیدا نمی کند و تفاوتی که دارند در قالب مقایسه قبل اغراض است ولی rbf SVM بر روی داده های train overfit می کند و دقت ۱۰۰ درصد را به ما ارایه می دهد که بهترین دقت در میان چهار مدل train شده است.

حال که هر چهار مدل را بر روی داده های train بررسی کردیم باید دست آورده هر کدام را بر روی داده های test مورد بررسی قرار دهیم تا بینیم میزان اعتمادی که به هر مدل می توان کرد در چه اندازه ای خواهد بود. برای شروع می توان گفت که نسبت به بخش قبل به طور کلی SVM نتایج به مراتب ضعیف تری در اختیار ما قرار خواهد داد.

بدترین نتیجه همان طور که در بند قبل گفتیم با مشاهده overfit در مدل rbf SVM به این مدل اختصاص پیدا می کند و دقت بسیار پایینی بر روی داده های test در اختیار ما قرار می دهد.

از بین سه مدل باقی مانده می توان گفت که هر سه تای آن ها یعنی SVM خطی SVM و SVM خطی OvO دقت تقریبا برابری بر روی داده های تست از خود نشان می دهند نسبت به یکدیگر برتری نشان نمی دهند هر چند که کماکان دقت خوبی در اختیار ما قرار نمی دهند و در یک نتیجه گیری کلی تمامی مدل های SVM train شده در این گزارش از MLP عملکرد بدتری بر روی داده های تست به نمایش در می آورند.

حال برای فهم بهتر از خروجی چهار مدل خود می توانیم از ماتریس برآنگی استفاده کنیم که نتایج بهتر را برای فهم قضیه در اختیار ما قرار می دهد.



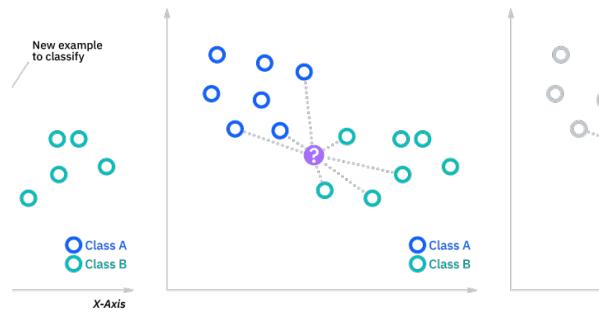
شکل ۲۹: ماتریس پراکندگی SVM ها

Knn (۳-۴)

الگوریتم K-Nearest Neighbors (KNN) همچنین به عنوان k-NN یا شناخته می‌شود، یک طبقه‌بندی کننده یادگیری non-supervised و parametric است که از تقریب نزدیکی نقاط برای انجام طبقه‌بندی یا پیش‌بینی در مورد گروه بندی یک نقطه داده فردی استفاده می‌کند.

در حالی که می‌توان از آن برای مسائل رگرسیون یا طبقه‌بندی استفاده کرد، معمولاً به عنوان یک الگوریتم طبقه‌بندی استفاده می‌شود، با این فرض که نقاط مشابه را می‌توان در نزدیکی یکدیگر یافت.

برای مسائل طبقه‌بندی، یک label کلاس بر اساس رای اکثریت اختصاص داده می‌شود به عبارتی برچسبی (label) که بیشتر از بقیه در اطراف یک نقطه داده شده معین نشان داده می‌شود مورد استفاده قرار می‌گیرد. در حالی که این از نظر فنی "رای گیری کثرت" در نظر گرفته می‌شود، اصطلاح "رای اکثریت" بیشتر در ادبیات استفاده می‌شود.



شکل ۳۰: روش محاسباتی kNN

حال ما مدل kNN خود را با تعداد همسایگی های ۵ پیاده سازی می کنیم.

دقت مدل kNN ما بر روی داده های train به صورت جدول زیر به دست می آید:

همچنین دقต مدل kNN ما بر روی داده های test به صورتی زیر به دست می آید:

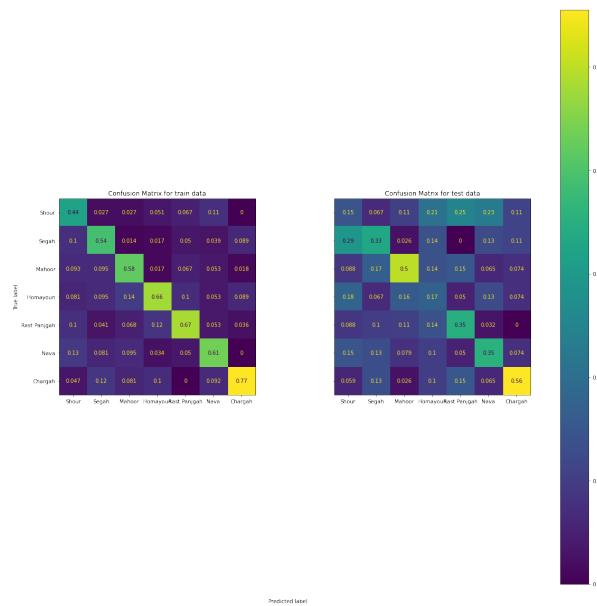
support	f1-score	recall	precision	
86	0.53	0.44	0.67	شور
74	0.59	0.54	0.65	سد گاه
74	0.61	0.58	0.63	ماهور
59	0.57	0.66	0.50	همایون
60	0.62	0.67	0.57	راست پنج گاه
76	0.61	0.61	0.61	نوا
56	0.66	0.77	0.57	چهار گاه
				Accuracy
485	0.60			macro avg
485	0.60	0.61	0.60	weighted avg
485	0.59	0.60	0.61	

جدول ۱۱: نتایج Knn بر روی دیتای train

support	f1-score	recall	precision	
34	0.15	0.15	0.16	شور
30	0.32	0.33	0.31	سه گاه
38	0.50	0.50	0.50	ماهور
29	0.18	0.17	0.19	همایون
20	0.33	0.35	0.32	راست پنج گاه
31	0.37	0.35	0.38	نوا
27	0.53	0.56	0.50	چهار گاه
209	0.34			Accuracy
209	0.34	0.34	0.34	macro avg
209	0.34	0.34	0.34	weighted avg

جدول ۱۲: نتایج Knn بر روی دیتای test

همان طور که مشاهده می شود دقت نهایی این مدل knn نسبت به SVM ها بهتر است و در حدود ۳۴ درصد قرار می گیرد و نتایج بهتری را در رابطه با طبقه بندی در اختیار ما قرار می دهد که در جدول بالا کاملا قبل مشاهده است. همچنین برای درک بهتر این مدل ماتریس پراکندگی آن به صورت زیر به دست می آید.



شکل ۳۱: ماتریس پراکندگی مدل knn

Logistic Regression (۴-۴)

به طور کلی Logistic Regression یک روش تحلیل آماری برای پیش‌بینی یک نتیجه باینری، مانند بله یا خیر، بر اساس مشاهدات قبلی یک مجموعه داده است. یک مدل Logistic Regression با تجزیه و تحلیل رابطه بین یک یا چند متغیر مستقل موجود، یک متغیر داده وابسته را پیش‌بینی می‌کند. در عمل Logistic Regression به یک ابزار مهم در رشته یادگیری ماشین تبدیل شده است. Logistic Regression به الگوریتم‌های مورد استفاده در برنامه‌های یادگیری ماشین اجازه می‌دهد تا داده‌های دریافتی را بر اساس داده‌های قبلی طبقه‌بندی کنند. با ورود داده‌های مرتبط اضافی، الگوریتم‌ها در پیش‌بینی طبقه‌بندی در مجموعه داده‌ها بهتر می‌شوند. مانند طبقه‌بندی کننده‌های قبل اینبار مدل Logistic Regression بر روی داده‌های خود پیاده سازی می‌کنیم.

بعد از پیاده سازی مدل نتایج آن بر روی داده‌های train به صورت زیر گزارش می‌شود:

support	f1-score	recall	precision	
42	0.36	0.43	0.32	شور
65	0.50	0.49	0.52	سه گاه
65	0.45	0.46	0.44	ماهور
88	0.53	0.50	0.56	هساپون
55	0.42	0.47	0.37	راست پنج گاه
85	0.54	0.51	0.57	نوا
85	0.54	0.51	0.57	چهار گاه
485	0.49			Accuracy
485	0.48	0.48	0.48	macro avg
485	0.49	0.49	0.50	weighted avg

جدول ۱۳: نتایج Logistic Regression بر روی دیتای train

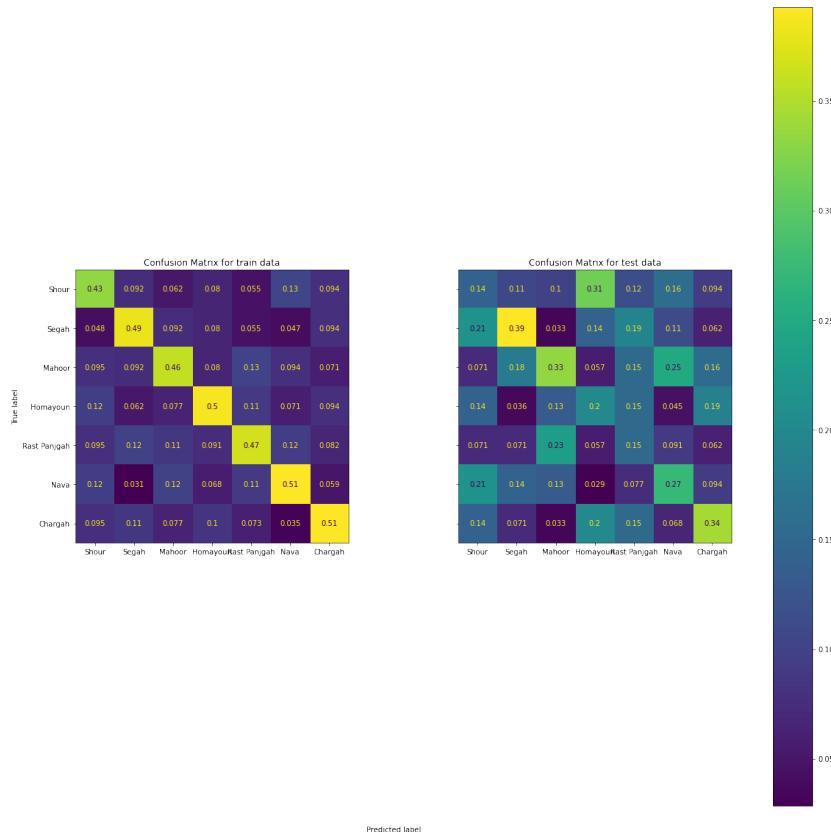
در ادامه همانند قبل مدل را بر روی داده‌های test پیاده سازی می‌کنیم نتیجه محکم تری از مدل طراحی شده خود به دست آوریم:

support	f1-score	recall	precision	
14	0.09	0.14	0.06	شور
28	0.37	0.39	0.34	سه گاه
30	0.29	0.33	0.26	ماهور
35	0.23	0.20	0.27	هساپون
26	0.17	0.15	0.18	راست پنج گاه
44	0.33	0.27	0.41	نوا
32	0.35	0.34	0.37	چهار گاه
209	0.27			Accuracy
209	0.26	0.26	0.27	macro avg
209	0.28	0.27	0.30	weighted avg

جدول ۱۴: نتایج Logistic Regression بر روی دیتای train

همان طور که از جدول بالا مشاهده می شود دقت مدل Logistic Regression بر روی داده های test در مقایسه با دو طبقه بندی کننده قبلی پایین تر است.

حال برای درک بهتر مدل به دست آمده Logistic Regression ماتریس پراکنندگی آن را به صورت زیر به دست می آوریم:



شکل ۳۲: ماتریس پراکنندگی Logistic Regression

Ensemble Learning (۵-۴)

یادگیری گروهی Ensemble Learning یک رویکرد خوب برای یادگیری ماشینی است که با ترکیب پیش‌بینی‌های چند مدل به دنبال عملکرد پیش‌بینی بهتری است.

اگرچه تعداد به ظاهر نامحدودی از مجموعه‌ها وجود دارد که می‌توانید برای مشکل مدل‌سازی پیش‌بینی‌کننده خود ایجاد کنید، سه روش وجود دارد که بر حوزه یادگیری گروهی Ensemble Learning تسلط دارند. به حدی که به جای الگوریتم‌ها فی نفسه، هر یک رشته‌ای هستند که روش‌های تخصصی‌تری را ایجاد کرده‌اند.

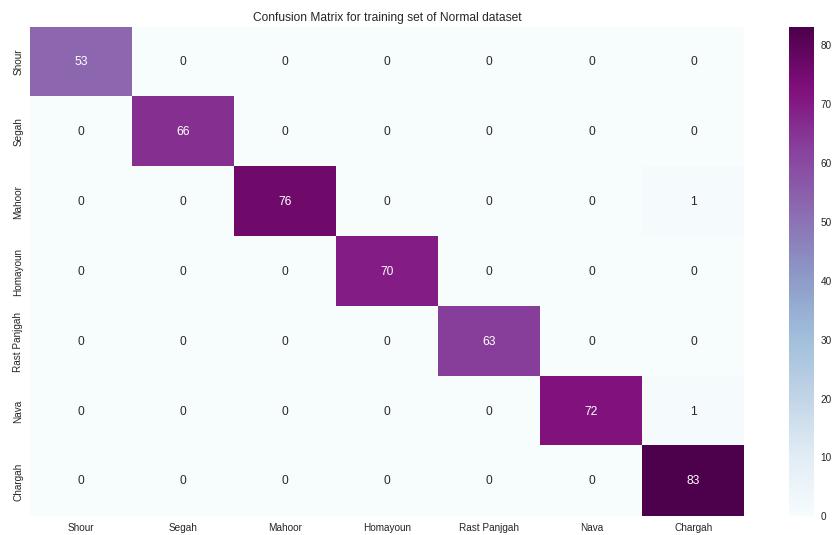
او چهار کلاس اصلی از روش‌های یادگیری گروهی عبارتند از: bagging, voting, stacking, boosting و مهم است که هم درک دقیقی از هر روش داشته باشید و هم آنها را در پروژه مدل‌سازی پیش‌بینی خود در نظر بگیرید.

اما، قبل از لایه‌بندی روی ریاضیات و کد، نیاز به مقدمه‌ای مختصر با این رویکردها و ایده‌های کلیدی پشت هر روش دارد. bagging شامل برآش بسیاری از درختان تصمیم بر روی نمونه‌های مختلف از یک مجموعه داده و میانگین گیری پیش‌بینی‌ها است. stacking شامل برآش انواع مدل‌های مختلف بر روی داده‌های مشابه و استفاده از مدل دیگری برای یادگیری نحوه ترکیب بهترین پیش‌بینی‌ها است.

boosting شامل افزودن متوالی اعضای گروه است که پیش‌بینی‌های مدل‌های قبلی را تصحیح می‌کنند و میانگین وزنی پیش‌بینی‌ها را به دست می‌آورند. voting شامل جمع کردن پیش‌بینی‌های انجام شده توسط مدل‌های طبقه‌بندی یا میانگین گیری پیش‌بینی‌های انجام شده توسط مدل‌های رگرسیون است. حال بعد از پیاده‌سازی مدل خود نتایج به صورت زیر تعریف می‌شود.

نتایج مدل ما بر داده‌ها نرم train به صورت زیر به دست می‌آید:

حال جدول نتایج به صورت زیر به دست می‌آید:



شکل ۳۳: نتایج Ensemble Learning بر روی داده‌های train

نتایج مدل ما بر داده‌ها نرمال test به صورت زیر به دست می‌آید:

حال جدول نتایج به صورت زیر به دست می‌آید:

نتایج مدل ما بر داده‌ها غیر نرمال train به صورت زیر به دست می‌آید:

حال جدول نتایج به صورت زیر به دست می‌آید:

support	f1-score	recall	precision	
53	1	1	1	شور
66	1	1	1	سه گاه
77	0.993	0.987	1	ماهور
70	1	1	1	همایون
63	1	1	1	راست پنج گاه
73	0.993	0.986	1	نوا
83	0.988	1	0.976	چهار گاه
485	0.996			Accuracy
485	0.996	0.996	0.997	macro avg
485	0.996	0.996	0.996	weighted avg

جدول ۱۵: نتایج Ensemble Learning بر روی داده های train

support	f1-score	recall	precision	
36	0.333	0.25	0.500	شور
28	0.419	0.464	0.382	سه گاه
29	0.295	0.310	0.281	ماهور
34	0.316	0.265	0.391	همایون
29	0.314	0.276	0.364	راست پنج گاه
31	0.521	0.613	0.452	نوا
22	0.467	0.636	0.368	چهار گاه
209	0.388			Accuracy
209	0.381	0.402	0.391	macro avg
209	0.376	0.388	0.396	weighted avg

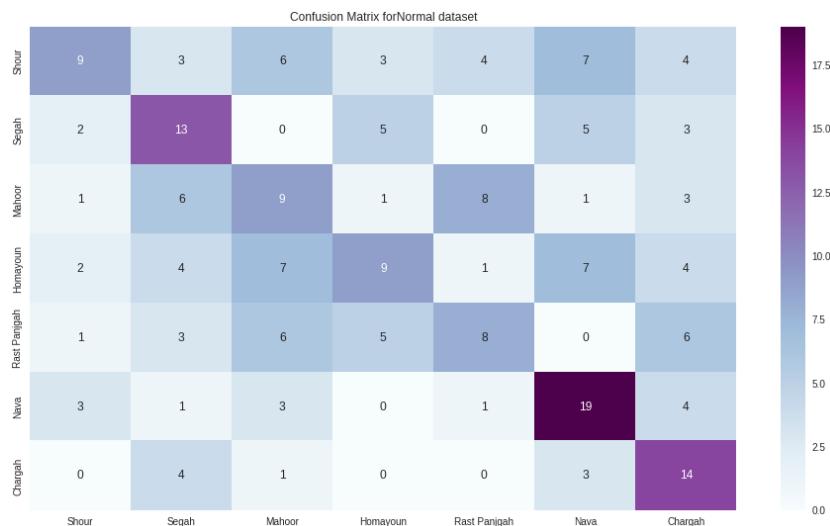
جدول ۱۶: نتایج Ensemble Learning بر روی داده های test

نتایج مدل ما بر داده ها غیر نرمال train به صورت زیر به دست می آید:

حال جدول نتایج به صورت زیر به دست می آید:

XGboost (۶-۴)

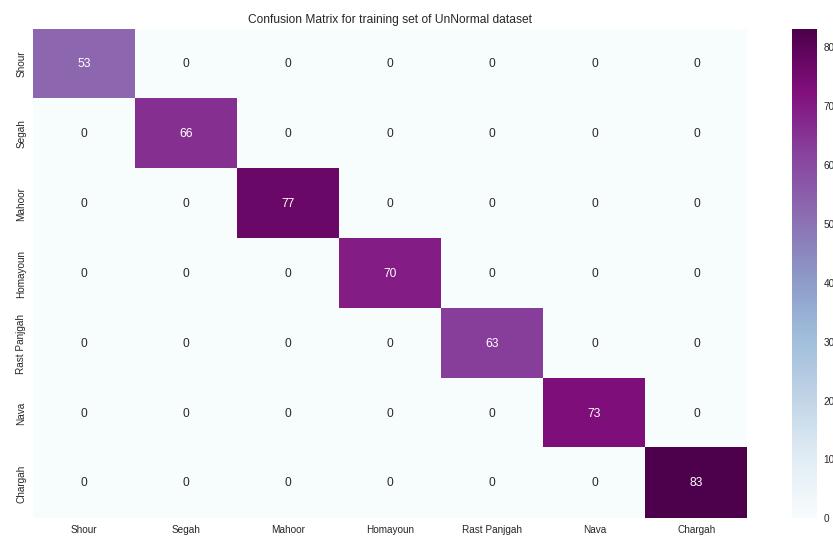
یک طبقه بند از دسته‌های learning ensemble XGBoost است که به طور کلی از نرکیب چندین و چند مدل ضعیفتر و قرار دادن آنها به صورت موازی یا سری به یک طبقه‌بند قویتر میرسد. یک درخت تصمیم که واحدهای سازنده‌ی جنگلهای تصادفی هستند در هر مرحله از عمقشان فضا را با استفاده از تنها یک ویزگی به دو قسمت تقسیم می‌کنند و با عمق بیشتر افزایهای 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.



شکل ۳۴: نتایج Ensemble Learning بر روی داده های test

اما سازندگان الگوریتم ماشین‌نینگ با استفاده از اصل خرد جمعی(با رایگیری از تعداد افراد بسیار زیاد خطأ 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 علاوه بر درختها به صورت مستقل نگاه میشود. روش‌های boosting برای حل این مشکل به وجود آمدند. در روش‌های boosting درختها به جای این که در توازن یکدیگر باشند با یکدیگر سری هستند و هر درخت روی خطأ درخت قبلی آموزش میبینند. روش‌های مبتنی بر گرادیان به این نحو عمل میکنند که درختهای بعدی همچلت 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 پیشینی میشد که مدل به داده‌های آموزش چسبنده شود که نتایج این نکته را تایید میکرد اما نکته 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 گفته هایی ما را تایید میکند، با توجه به اینکه Score Validation -Cross Validation همواره با افزایش نمونه‌ها در حال افزایش است، میتوان نتیجه گیری کرد که مدل تشنگ داده است. برای درک بهتر باید مد نظر داشته باشیم دو منبع خطأ به طور کلی وجود دارد: الف) اشتباه در فرض اولیه برای مدل b (کم بودن تعداد داده‌ها برای ابعاد مسئله حال با توجه به نمودارهای شکل ۱۴ و دقتی که برای مدل به دست آورده‌یم می‌توان نتیجه گرفت که در مدل ساخته خطأی "ب" جلوگیری میکند از عملکرد بهتر مدل. در نهایت عملکرد مدل را نیز میتوان مشاهده کرد به ازای فیچر‌های مختلف: با توجه نتایج مشاهده میکنیم که در دو کلاس لری و بندری با توجه به ریتمی که دارند خطای کمتری وجود دارد اما در کلاس کردي، با توجه به این نکته که آهنگ‌هایي کردي از لحاظ ریتم و موسیقی همه طیف‌ها را در خود دارد در نتیجه بیشترین خطأ را دارد. اما در مورد خطأ متوجه میشویم که به داده‌های آموزش fit over شده است و برای حل مشکل دو راه وجود دارد، استفاده از مدل ساده تر و افزایش نمونه‌ها. در نمودار score val cross و توضیحات آن گفته شد که اضافه کردن داده به مدل میتواند راه حل مناسب تری باشد.

حال مدل ما بر روی داده‌های نرمال train به صورت زیر خواهد بود:



شکل ۳۵: نتایج Ensemble Learning train بر روی داده های غیر نرمال

و جدول آن به صورت زیر خواهد بود:

در ادامه مدل ما بر روی داده های نرمال test به صورت زیر خواهد بود:

(۷-۴) مقایسه

داده ها پیچیده بودن این نوع داده، همانطور که پیشینی می شد مدل های مانند SVM خطی و به طور کلی مدل های ساده، دقت نسبتاً پایینی برای طبقه بندی می دهند. بهترین مدل در این بخش مدل شبکه عصبی بود که مشخص است با توجه به اینکه این مدل قابلیت استخراج روابط پیچیده است قابل پیش بینی بود. همچنین شایان ذکر است که در تمام مدل ها استفاده از داده نورمال نشده، باعث بدتر شدن مدل ها می شود.

(۵) خوش بندی

خوش بندی یا به عبارتی Clustering وظیفه گروه بندی مجموعه ای از اشیاء است به گونه ای که اشیاء در یک گروه (به نام خوش) شباهت بیشتری به یکدیگر داشته باشند تا در گروه های دیگر و به طور کلی با روش های مختلف داده ها به خوش های مختلف تقسیم می شوند.

support	f1-score	recall	precision	
26	0.280	0.269	0.292	شور
27	0.339	0.370	0.312	سه گاه
36	0.349	0.306	0.407	ماهور
31	0.406	0.452	0.368	همایون
25	0.310	0.360	0.273	راست پنج گاه
38	0.364	0.316	0.429	نوا
26	0.415	0.423	0.407	چهار گاه
209	0.354			Accuracy
209	0.352	0.357	0.356	macro avg
209	0.354	0.354	0.363	weighted avg

جدول ۱۷: نتایج Ensemble Learning بر روی داده های غیر نرمال train

support	f1-score	recall	precision	
65	1	1	1	شور
59	1	1	1	سه گاه
78	1	1	1	ماهور
80	1	1	1	همایون
64	1	1	1	راست پنج گاه
73	1	1	1	نوا
71	1	1	1	چهار گاه
209	1			Accuracy
209	1	1	1	macro avg
209	1	1	1	weighted avg

جدول ۱۸: نتایج Ensemble Learning بر روی داده های غیر نرمال test

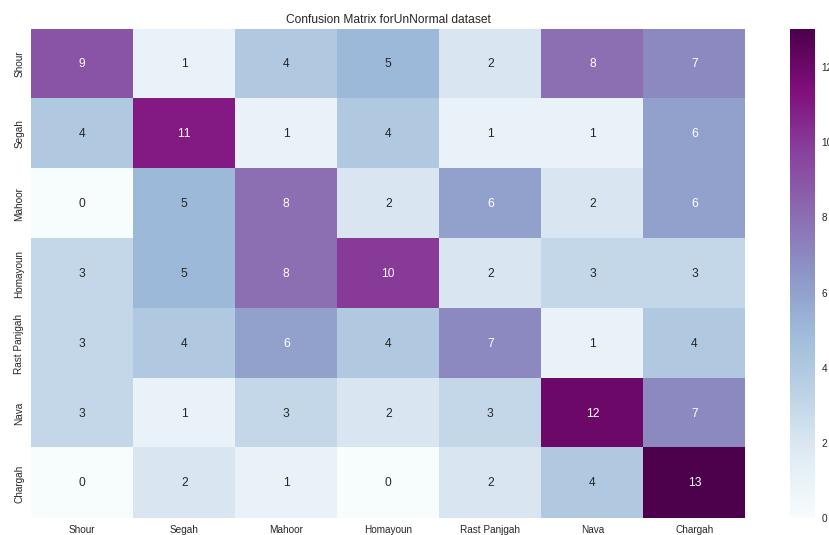
K-means (۱-۵)

به طورکلی در K-means هدف این است که مرکزهایی را انتخاب کند که اینرسی، یا مجموع مربعات درون خوشه را به حداقل برساند.

$$\sum_{i=0}^n \min_{\mu_j \in C} (\|x_i - \mu_j\|^2) \quad (9)$$

اینرسی را می توان به عنوان معیاری برای میزان انسجام درونی خوشه ها درنظر گرفت. از معایب مختلفی تاثیر می پذیرد:

- ۱) اینرسی این فرض را ایجاد می کند که خوشه ها محدب و همسانگرد هستند، که همیشه اینطور نیست. به خوشه های دراز یا منیفوولد هایی با اشکال نامنظم واکنش ضعیفی نشان می دهد.
- ۲) اینرسی یک متريک نرمال شده نیست: ما فقط می دانیم که مقادير کمتر بهتر است و صفر بهينه خواهد بود. اما در فضاهای با ابعاد بسيار بالا، فواصل اقليليسی تمایل دارند که متورم شوند (این نمونه ای از به اصطلاح "نفرین ابعاد" است). اجرای یک الگوريتم کاهش



شکل ۳۶: نتایج Ensemble Learning بر روی داده های غیر نرمال test

بعاد مانند تجزیه و تحلیل مؤلفه اصلی (PCA) قبل از خوشه بندی k-means می تواند این مشکل را کاهش دهد و محاسبات را سرعت بخشد.

در پیاده سازی K-means سه مرحله دارد:

-اولین مرحله، مرکزهای اولیه را انتخاب می کند، با اساسی ترین روش، انتخاب K نمونه از مجموعه داده ها

-پس از مقداردهی اولیه، K-means شامل حلقه زدن بین دو مرحله دیگر است.

(۱) مرحله اول هر نمونه را به نزدیکترین مرکز خود اختصاص می دهد.

(۲) مرحله دوم با در نظر گرفتن مقدار مبانگین تمام نمونه های اختصاص داده شده به هر مرکز قبلي، مرکزهای جدید ایجاد می کند. تفاوت بین مرکز قدیمی و جدید محاسبه می شود و الگوریتم این دو مرحله آخر را تا زمانی که این مقدار کمتر از یک آستانه باشد تکرار می کند. به عبارت دیگر، آنقدر تکرار می شود که مرکزها حرکت قابل توجهی نداشته باشند.

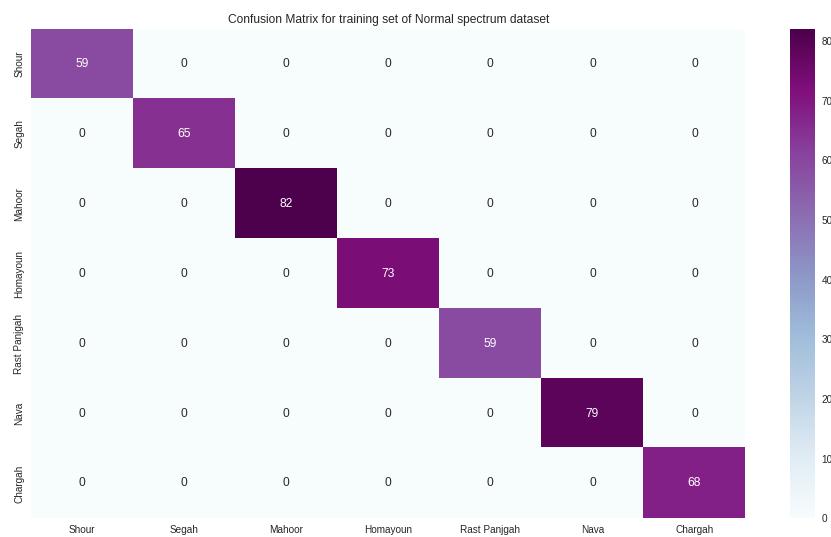
برای راحتی درک این خوشه بندی برای دو تا از PCA ها مهمتر را نیز جدا کرده ایم و شکل زیر را رسم کردیم. مشخص است که داده ها به سادگی جدایی پذیر نیستند و خوشه بندی برای حالات ۲ و ۵ و ۷ به شکل زیر می باشد:

حال با توجه به ماتریس های در هر یختنگی موارد زیر را می توان استنتاج کرد:

- با توجه به کلاس بندی دو کلاسه فرضیه ما راجح به سادگی جدا پذیر نبودن داده ها تایید می شود. یعنی مشخص است که بیشتر داده ها در یک خوشه قرار گرفته اند.

- کلاس بندی ۷ کلاسه نیز تا حد زیادی داده ها به صورت تصادفی پخش شده اند، این قضیه نیز با توجه به ماهیت داده ها قابل توجیه است زیرا که با توجه به توزیع داده ها و شبیه بودن این داده ها به هم، مشخص است که خوشه بندی در بعد فعلی امکان پذیر نبوده و نتایج صحیح می باشد.

- باتوجه به خوشه بندی ۲۱ تایی همان طور که انتظار می رفت داده ها به صورت تصادفی پخش شده اند این قضیه با توجه به این



شکل ۳۷: نتایج XGboost بر روی داده های غیر نرمال train

که توضیع داده ها بسیار شبیه به هم بود در بعد فعلی جدایی پذیر نیست قابل توجیه است. همچنین ماتریس آشفتگی می توان از شباهت بین کلاس و خوشه ها پی برد برای مثال بیشتر داده های کلاس پنج در خوشه هفتم قرار گرفته است همچنین تعداد زیادی از داده های کلاس یک، دو، سه و هفت در یک خوشه قرار گرفته اند که به معنی شبیه بودن این کلاس ها به یکدیگر است.

البته باید دقت داشت تعداد خوشه های مناسب باید با استفاده از روش زانویی سنجیده شود. با توجه به نمودار زانویی رسم شده، نیز می توان فهمید بهترین تعداد خوشه ۷ تا است که به تعداد کلاس هایمان می باشد.

Spectral Clustering (۲-۵)

خوشه بندی طیفی را می توان به عنوان خوشه بندی نموداری در نظر گرفت. برای داده های مکانی می توان به القای یک نمودار بر اساس فواصل بین نقاط (به طور بالقوه یک نمودار k-NN یا حتی یک نمودار متراکم) فکر کرد. از آنجا، خوشه بندی طیفی به بردارهای ویژه لپلاسین گراف نگاه می کند تا تلاش کند یک جاسازی خوب (بعد پایین) نموداری در فضای اقلیدسی پیدا کند. این اساساً نوعی یادگیری چندگانه است، یافتن دگرگونی فضای اصلی به گونه ای است که فاصله های چندگانه را برای چند منیفولی که داده ها فرض می شود در نظر می گیرد، بهتر نشان دهد. هنگامی که فضای تبدیل شده را داشته باشیم، یک الگوریتم خوشه بندی استاندارد اجرا می شود. با 'sklearn' پیش فرض K-Means است. این بدان معناست که کلید خوشه بندی Spectral، دگرگونی فضا است. با فرض اینکه بهتر بتوانیم manifold را اجرا کنیم، خوشه بندی بهتری به دست خواهیم آورد. ما نیاز به نگرانی کمتری در مورد خوشه های کروی K-Means داریم، زیرا آنها فقط در فضای تبدیل شده کروی هستند و نه فضای اصلی. ما متأسفانه برخی از نقاط ضعف K-Means را حفظ می کنیم: ما همچنان داده ها را به جای خوشه بندی آن ها تقسیم بندی می کنیم. حدس زدن پارامتر "تعداد خوشه ها" سخت است. ما مشکلات پایداری داریم که از K-Means به ارث رسیده است. بدتر از آن، اگر بر روی نمودار متراکم ماتریس فاصله عمل کنیم، گام اولیه بسیار گرانی خواهیم داشت و عملکرد را قربانی می کنیم.

support	f1-score	recall	precision	
65	1	1	1	شور
59	1	1	1	سه گاه
78	1	1	1	ماهور
80	1	1	1	همایون
64	1	1	1	راست پنج گاه
73	1	1	1	نوا
71	1	1	1	چهار گاه
				Accuracy
485	1			macro avg
485	1	1	1	weighted avg

جدول ۱۹: نتایج XGboost بر روی داده های غیر نرمال test

در کل می توان گفت که با توجه به این روش به این که این روش تمرکز زیادی رو نقاط اطراف دارد و نزدیک بودن مقادیر ویژگی ها در این روش به هم، اکثر داده ها در یک کلاس قرار گرفته و با توجه به این موضوع روش مناسبی نخواهد بود. و البته می توان با توجه به ماتریس در هم ریختگی گفت که داده ها چگال هستند با فاصله زیاد از هم و در دو خوش دسته بندی می شوند.

(۶) نتیجه گیری

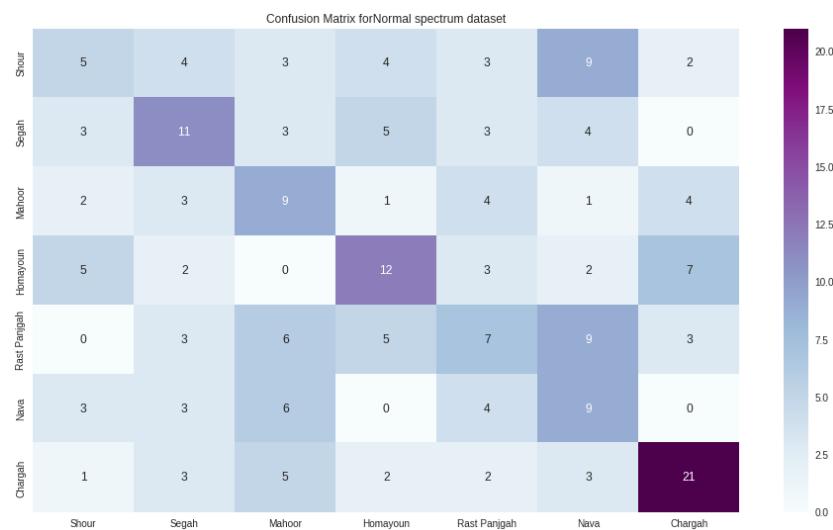
مقایسه بین مدل ها و نتیجه گیری

مقایسه کلی

با توجه به نوع داده ها و اینکه داده های صوتی به سادگی جدایی پذیر نیست، دقت پایین اکثر مدل ها توجیه پذیر می باشد. بهترین مدل ها برای اینکار مدل هایی بود که می توانست داده را به فضایی دیگر تصویر کند و در فضای جدید داده ها را طبقه بندی کند. برای مثال طبقه بندهای MLP و RBF SVM بهترین نتیجه ها را در مقایسه با سایر مدل ها می دهند. شایان به ذکر است که همانطور که در بخش های قبلی بیشنهاد شد، برای داده های صوتی بهتر است از مدل های حافظه دار مانند LSTM استفاده شود. همچنین در مواردی مانند نورمالایز شده و نشده

نورمال سازی یک تکنیک مقیاس پذیری در یادگیری ماشینی است که در طول آماده سازی داده ها برای تغییر مقادیر ستون های عددی در مجموعه داده به منظور استفاده از یک مقیاس مشترک استفاده می شود. برای همه مجموعه داده ها در یک مدل ضروری نیست. تنها زمانی مورد نیاز است که ویژگی های مدل های یادگیری ماشین دارای محدوده های متفاوتی باشند. برای داده های ما با توجه به اینکه جنس ویژگی ها مختلف بوده، و همچنین در رنج های مختلف قرار دارند لازم است که حتما نورمالیزیشن صورت بگیرد.

نورمال کردن داده ها در اکثر موارد باعث بهتر شدن نتیجه نهایی همه مدل ها می شود. البته در بعضی از مدل ها مانند XGBoost به صورت built-in این موضوع پیاده سازی شده است. با مقایسه دقت و ماتریس های در هم ریختگی مدل های مختلف لزوم نورمالایزیشن واضح است.



شکل ۳۸: نتایج XGboost بر روی داده های غیر نرمال test

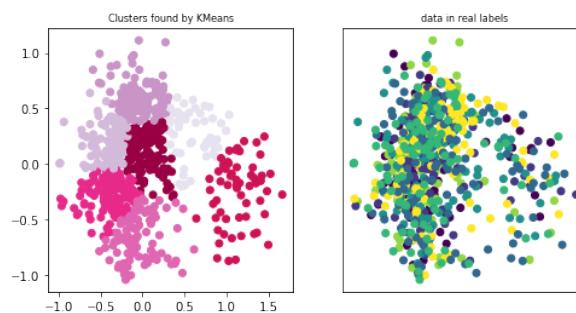


شکل ۳۹: دسته بندی های مختلف با kmeans

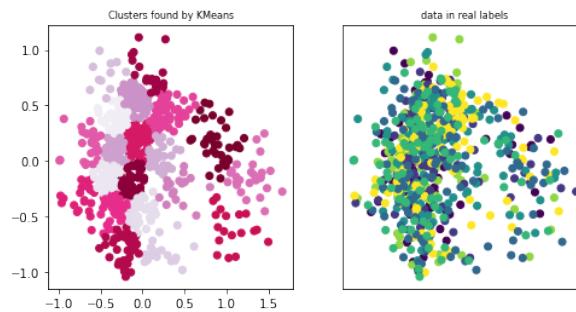
(۷) ضمیمه

کدهای پیاده سازی به صورت فایل های نوت بوک در کنار این پروژه پیوست شده اند. در فایل Feature extraction ابتدا، ویژگی های مختلف فایل های صوتی به نمایش در آمدند. همچنین، ویژگی های مختلفی استخراج شده اند و در فایل های CSV قرار گرفته اند. در این پروژه دو دسته بندی برای کارهای خوش بندی و طبقه بندی در نظر گرفته شده است که هر کدام در فایل های کد متناسب نتایج آن قابل مشاهده است.

همچنین برای داده های نورمالایز شده، نورمالایز شده و حالت کاهش ویژگی در فایل های مختلف عملیات های لازم انجام شده است و قابل مقایسه است. عملیات های طبقه بندی در فایل هایی با نام Classification قرار گرفته اند. همچنین عملیات های خوش بندی در فایل هایی با نام clustering قرار گرفته اند.



شکل ۴۰: دسته بندی های مختلف با kmeans



شکل ۴۱: دسته بندی های مختلف با kmeans

کاهش ویژگی

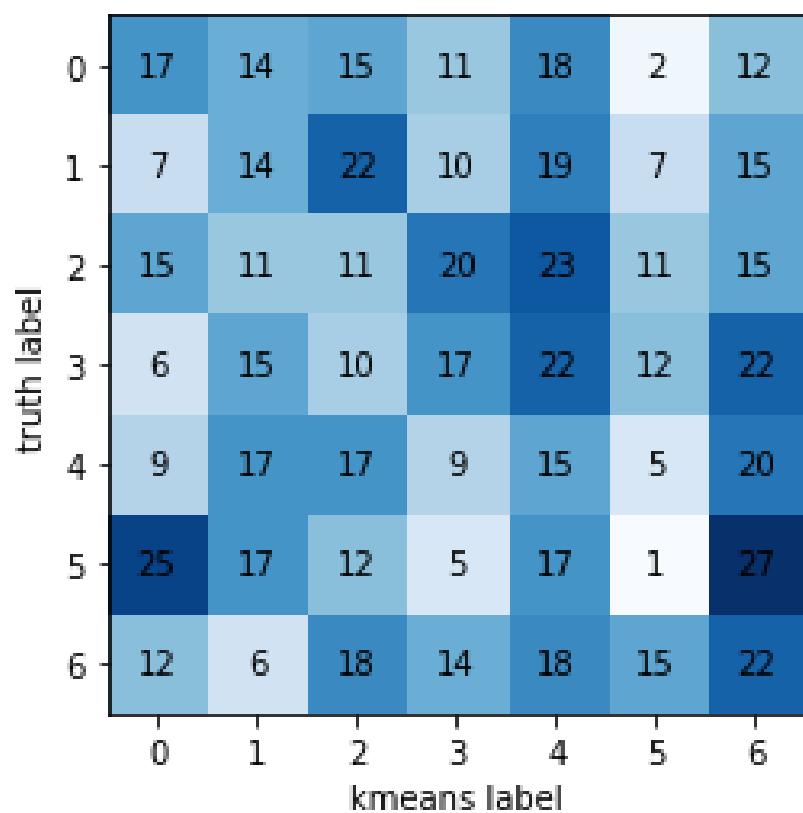
کاهش ویژگی با استفاده از روش های انتخاب ویژگی معمولاً منجر به بدتر شدن طبقه بندی های ما می شود. این موضوع به این دلیل است که ویژگی های استخراج شده هیچ کدام به تنها یابی قابلیت توجیه داده ها را ندارند. همچنین شایان به ذکر است که در مواردی مانند PCA این روش سعی دارد ویژگی هایی را نگه دارد که بیشترین واریانس را در آن ها داریم، لذا با توجه به اینکه این واریانس را می توان به نویز نیز نسبت داد. در شرایطی که داده های ما سیگنال صوتی هستند کاهش ویژگی می تواند باعث بدتر شدن نتیجه هی نهایی شود.

ویژگی های مختلف

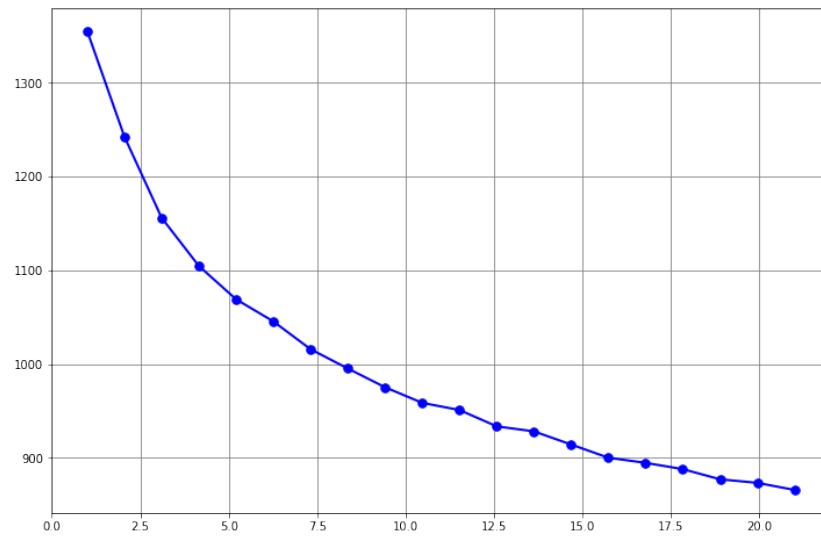
ما دو دسته از ویژگی های chroma و mfcc در کنار mel-spectrogram برای آموزش مدل ها استفاده کردیم. ویژگی spec-trogram به طور کلی نتیجه هی بدتری نسبت به دیگر ویژگی به ما می داد. البته این موضوع با توجه به ماهیت این ویژگی قابل بیان بود. برای هر کدام از این ویژگی های roc آن نمایش داده شده است که کامل ها آن ها در کد قابل رویت است.

خوبه بندی

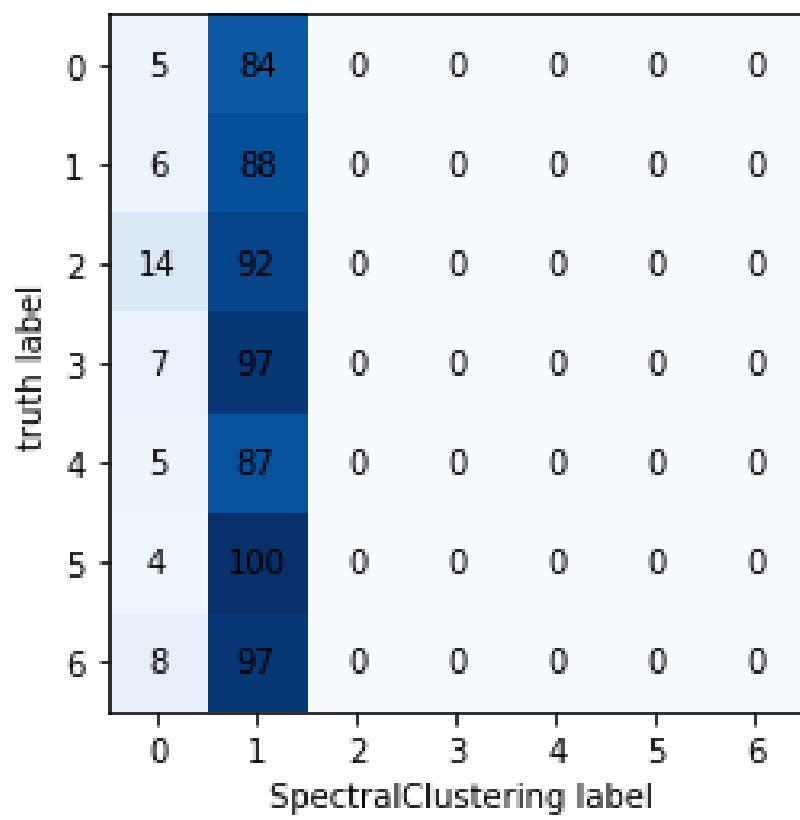
الگوریتم خوبه بندی kmeans با توجه به اینکه داده ها، در حالت عادی بسیار نزدیک به هم هستند قابلیت جداسازی مناسبی ندارد، اما با توجه به این الگوریتم، شباهت بین داده ها را می توانیم مشاهده کنیم.



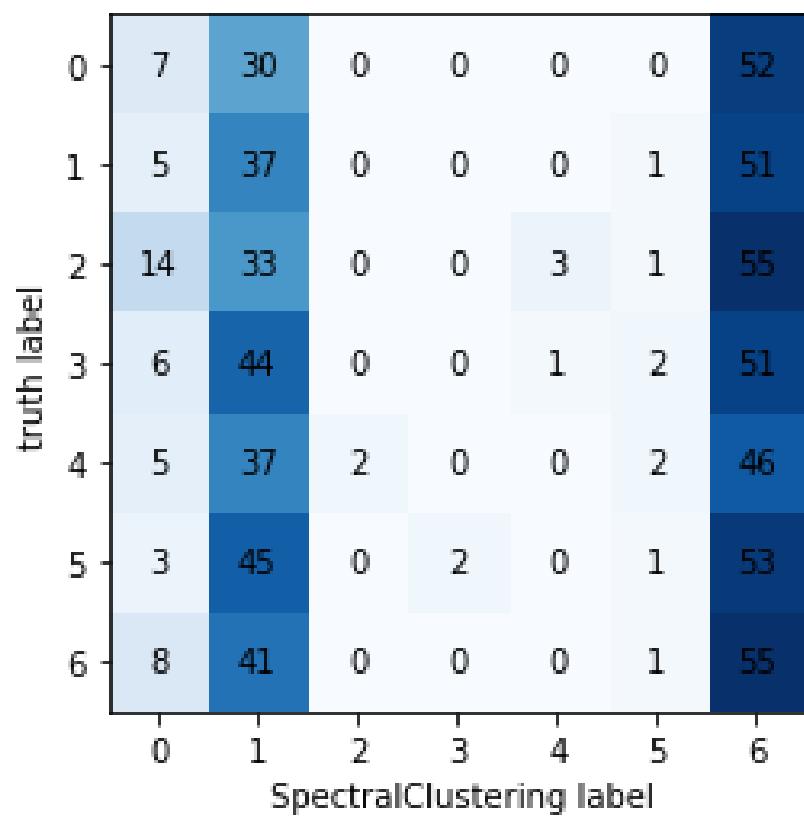
شکل ۴۲: خوشبندی K-means



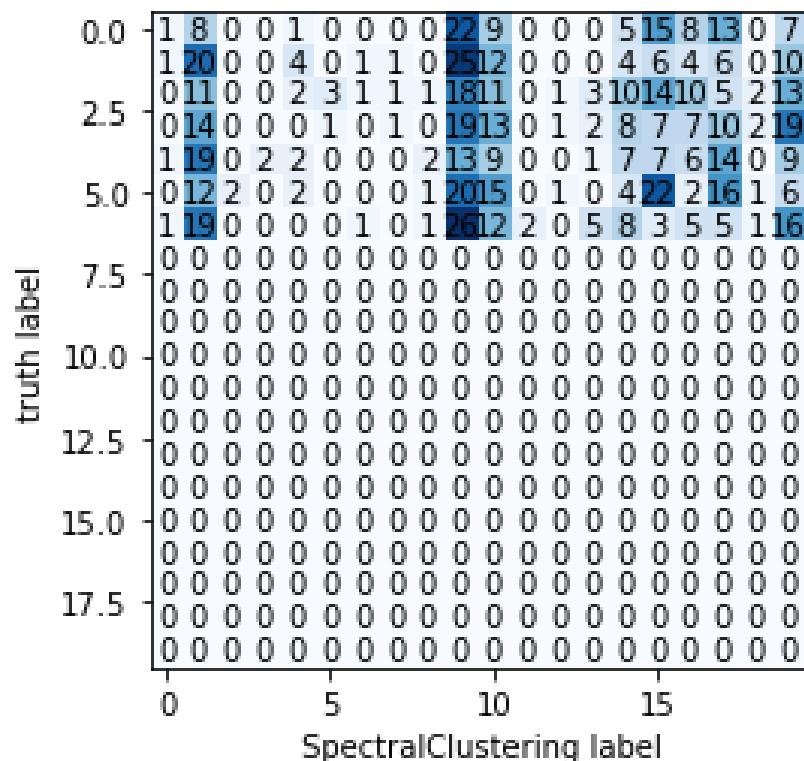
شکل ۴۳: نمودار زانویی K-means



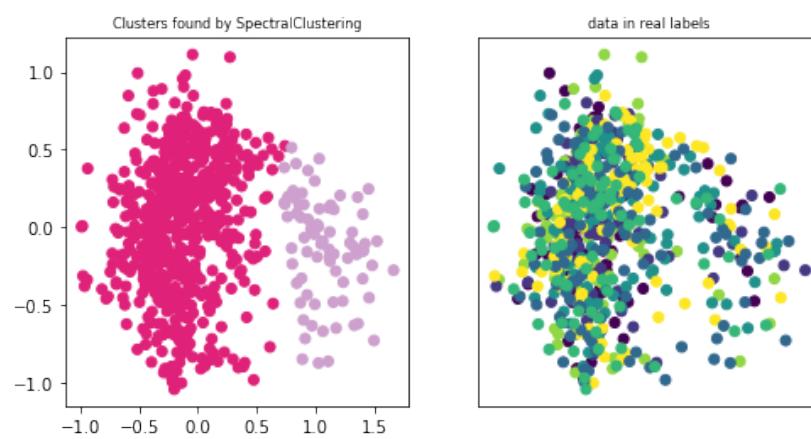
شکل ۴۴: ماتریس پاکندگی spectral



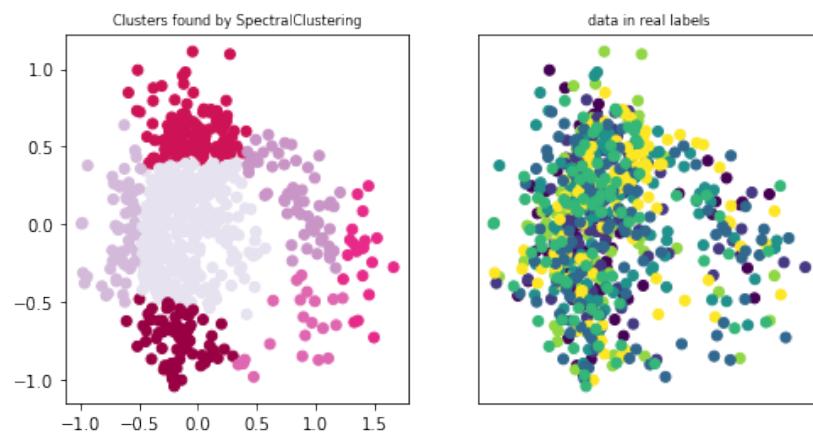
شکل ۴۵: ماتریس پاکندگی spectral



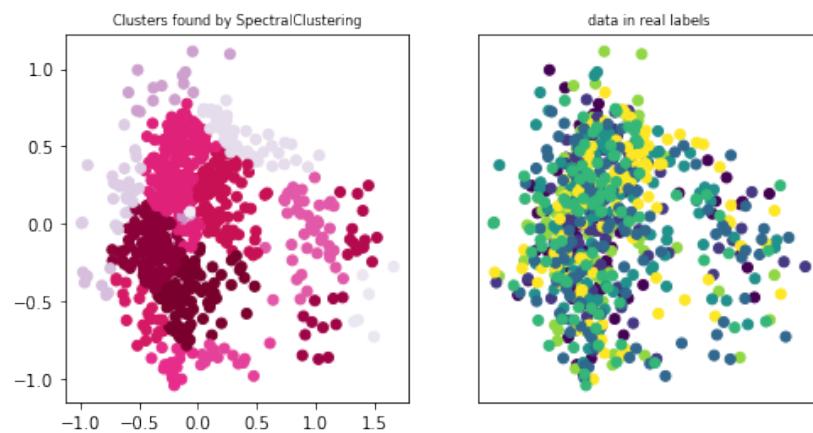
شکل ۴۶: ماتریس پاکندگی spectral



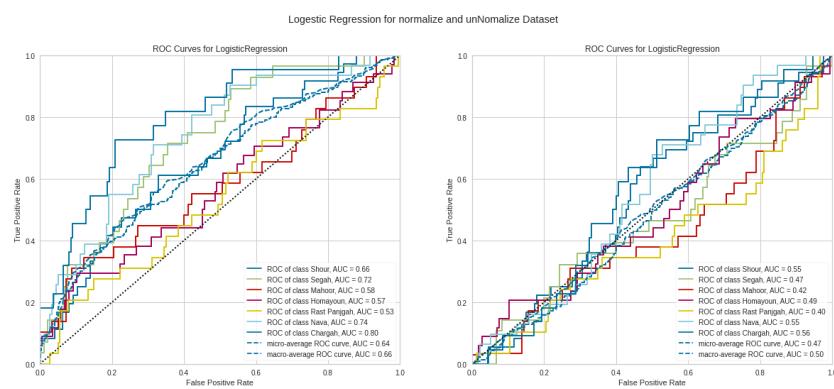
شکل ۴۷: خوشبندی spectral



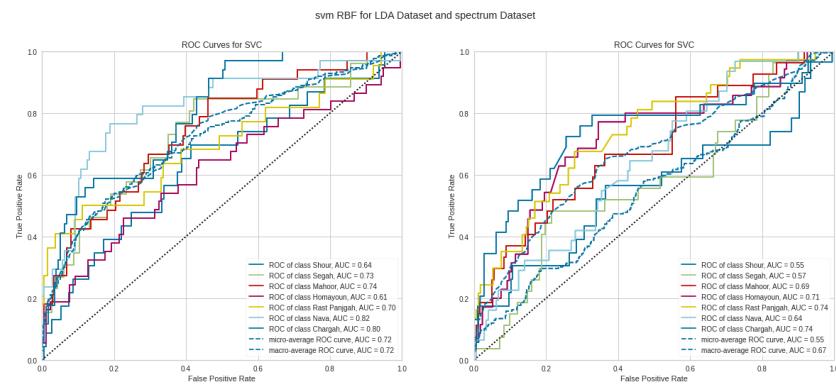
شکل ۴۸: خوشه بندی spectral



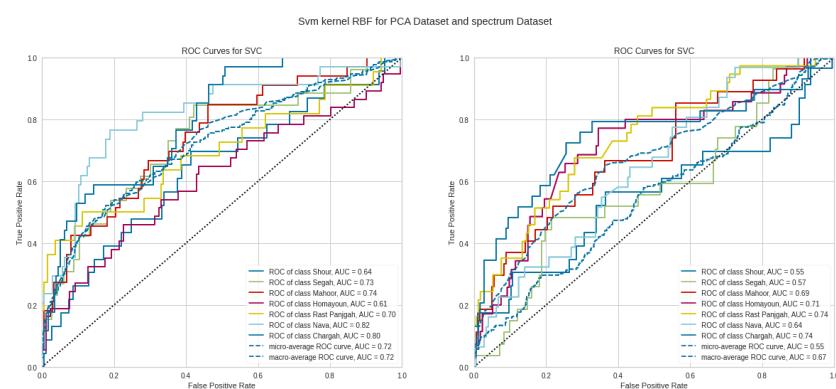
شکل ۴۹: خوشه بندی spectral



شکل ۵۰: نرمالایز شده و ننشده



شکل ۵۱: svm برای LDA



شکل ۵۲: svm برای PCA