



دانشگاه تهران

دانشکده برق و کامپیوتر

پروژه پایانی درس یادگیری ماشین

دانشجویان :

محمد امینی

شماره دانشجویی : ۸۱۰۱۰۰۲۹۴

محمد ناصری

شماره دانشجویی : ۸۱۰۱۰۰۴۸۶

فهرست مطالب

۴		۱	مقدمه
۵	۱.۱	هدف پژوهش
۶		۲	پیش پردازش داده ها
۶	۱.۲	معرفی دیتاست
۶	۲.۲	تمیزسازی دیتاست
۹	۳.۲	استخراج ویژگی
۹ MFCC (Mel Frequency Cepstral Coefficients)	۱.۲.۲	
۱۰ Chroma Frequencies	۲.۳.۲	
۱۰ جمع بندی کارهای انجام شده در این بخش	۴.۲	
۱۱		۳	طبقه بندی
۱۱	۱.۳	مقدمه
۱۱ K-Nearest Neighbor (KNN)	۲.۳	طبقه بند (KNN)
۱۳ اعمال KNN برای طبقه بندی "احساس" با استفاده از دسته ویژگی MFCC	۱.۲.۳	اعمال KNN برای طبقه بندی "احساس"
۱۵ اعمال KNN برای طبقه بندی "احساس" با استفاده از دسته ویژگی Chroma Frequency	۲.۲.۳	اعمال KNN برای طبقه بندی "احساس"
۱۷ طبقه بند MLP	۳.۳	طبقه بند MLP
۱۸ اعمال MLP برای طبقه بندی "احساس" با استفاده از دسته ویژگی MFCC	۱.۳.۳	اعمال MLP برای طبقه بندی "احساس"
۲۰ اعمال MLP برای طبقه بندی "احساس" با استفاده از دسته ویژگی Chroma Frequency	۲.۳.۳	اعمال MLP برای طبقه بندی "احساس"
۲۲ طبقه بند SVM	۴.۳	طبقه بند SVM
۲۴ اعمال SVM برای طبقه بندی "احساس" با استفاده از دسته ویژگی MFCC	۱.۴.۳	اعمال SVM برای طبقه بندی "احساس"
۲۵ اعمال SVM برای طبقه بندی "احساس" با استفاده از دسته ویژگی Chroma Frequency	۲.۴.۳	اعمال SVM برای طبقه بندی "احساس"
۲۹ اعمال KNN برای طبقه بندی "جنسیت" با استفاده از دسته ویژگی MFCC	۳.۴.۳	اعمال KNN برای طبقه بندی "جنسیت"
۳۰ اعمال KNN برای طبقه بندی "جنسیت" با استفاده از دسته ویژگی Chroma Frequency	۴.۴.۳	اعمال KNN برای طبقه بندی "جنسیت"
۳۱ اعمال MLP برای طبقه بندی "جنسیت" با استفاده از دسته ویژگی MFCC	۵.۴.۳	اعمال MLP برای طبقه بندی "جنسیت"
۳۲ اعمال MLP برای طبقه بندی "جنسیت" با استفاده از دسته ویژگی Chroma Frequency	۶.۴.۳	اعمال MLP برای طبقه بندی "جنسیت"
۳۳ اعمال SVM برای طبقه بندی "جنسیت" با استفاده از دسته ویژگی MFCC	۷.۴.۳	اعمال SVM برای طبقه بندی "جنسیت"
۳۴ اعمال MLP برای طبقه بندی "جنسیت" با استفاده از دسته ویژگی Chroma Frequency	۸.۴.۳	اعمال MLP برای طبقه بندی "جنسیت"

۴۷	خوشبندی	۴
۴۷	PCA	۱.۴
۴۸	t-SNE	۲.۴
۴۰	UMAP	۳.۴
۴۱	LDA	۴.۴
۴۲	معرفی روش‌های کلاسترینگ	۵.۴
۴۲	K-Means	۱۵.۴
۴۲	Agglomerative Hierarchical Clustering	۲۵.۴
۴۳	نتایج خوشبندی:	۶.۴
۴۳	2- Cluster	۱۶.۴
۴۴	4- Cluster	۲۶.۴
۴۵	10- Cluster	۳۶.۴
۴۶	جمع‌بندی نهایی :	۷.۴

فصل ۱

مقدمه

بروز احساس یکی از راههای بیان تفکر و منظور انسان‌هاست. یک فرد احساس خود را در قالب آرایه‌های گفتاری، تغییرات صورت، حرکات سر و بدن و سیگنل‌های زیستی مانند ضربان قلب و دمای نواحی خاص بدن بیان می‌کند.

تشخیص حس گفتار امری ضروری در ارتباط بین انسان و ماشین است. از جمله این کاربردها نظریه آموزش مجازی، ایجاد موتوهای جستجوی احساسی، برنامه‌های کامپیوتری، پیام‌های صوتی براساس حس گوینده و برنامه‌های کاربردی مشابه است. هم‌چنین گزارش شده است که در سیستم‌های تشخیص گفتار خلبان، استفاده از گفتارهای آموزشی همراه با استرس، عملکرد بهتری نسبت به گفتار معمولی دارد. در علم پژوهشی نیز می‌توان از سیستم تشخیص احساس از روی گفتار به عنوان ابزاری برای تشخیص بیماری‌های خاص مانند بیماری‌های قلبی و سرطان استفاده گردد.

یادگیری ماشین یا ماشین لرنینگ (Machine Learning) یکی از زیر مجموعه‌های هوش مصنوعی است که به سیستم‌ها این امکان را می‌دهد تا به صورت خودکار یادگیری و پیشرفت داشته باشند بدون اینکه به برنامه نویسی صریحی برای آن داشته باشند. تمرکز اصلی یادگیری ماشینی بر توسعه برنامه‌های رایانه‌ای است که بتوانند به داده‌ها دسترسی پیدا کنند و از آن برای یادگیری خود استفاده کنند. فرآیند یادگیری با مشاهدات یا داده‌ها آغاز می‌شود، مانند مثال‌ها، تجارت‌های مستقیم و یا دستور العمل‌ها، تا به یک الگو در داده‌ها برسند و بر اساس این مثال‌هایی که ارائه می‌دهیم، تصمیمات بهتری بگیرند. هدف اصلی آن است که به کامپیوتر این اجزاء را بدهیم که بدون دخالت و کمک انسان به طور اتوماتیک یادگیری داشته باشند و بتوانند اقدامات خود را بر مطابق با آن تنظیم کنند.

هدف اساسی الگوریتم‌های یادگیری ماشین، تعمیم یادگیری‌ها به فراتر از نمونه‌های آموزش داده شده است، یعنی تفسیر موفقیت آمیز داده‌ها. امروزه کاربرد یادگیری ماشین در زندگی روزمره انسان به صورت روزافزون در حال افزایش است، کاربردهای جذاب آن مانند:

- پردازش تصویر، بینایی کامپیوتری، تشخیص چهره، تشخیص حرکت و تشخیص شیء
- زیست‌شناسی محاسباتی، تشخیص تومورهای سرطانی، تعیین توالی DNA، کشف دارو و حتی موارد مخدوش
- تولید انرژی، پیش‌بینی قیمت بازار
- ساخت خودرو، حوزه هواپما
- پردازش زبان طبیعی، برنامه‌های تشخیص و شناسایی صدا و زبان
- و ...

۱.۱ هدف پروژه

در این پروژه ، هدف طبقه بندی(Clustering) و خوشه بندی(Classification) ۴ نوع احساس (عصبانیت ، ناراحتی ، خوشحالی و خنثی) از روی صدا است به طوری که بتوان پس از انجام پروژه با استفاده از بهترین روش ، با درصد خطای حداقلی و دقت حداکثری، بین این چهار نوع احساس تمایز قائل شد و با استفاده از الگوریتم های استفاده شده، این چهار نوع احساس را تشخیص داد و طبقه بندی کرد. برای این پروژه از الگوریتم های K-Means و SVM و MLP در بخش طبقه بندی (Classification) و از Agglomerative و K-Means در بخش خوشه بندی (Clustering) استفاده شده است.

فصل ۲

پیش پردازش داده‌ها

۱.۲ معرفی دیتاست

به فرآیند استخراج داده خام برای وظایف یادگیری ماشین، جمع آوری داده می‌گویند. این داده از روش‌های مختلفی مانند منابع آنلاین منبع باز یا منابع غیر رایگان، به دست می‌آید. شاید این مرحله را بتوان مهم‌ترین مرحله یادگیری ماشین نامید. اگر داده جمع آوری شده کیفیت پایینی داشته یا غیر مرتبط باشد، مدل آموزش دیده نیز کیفیت پایینی دارد. در این پژوهه از هر دانشجو به صورت انفرادی خواسته شده است که برای ۱۰ جمله مشخص شده، با احساسات: عصبانیت، ناراحتی، خوشحالی و خنثی از دو نفر آقا و دو نفر خانم صدا جمع آوری شود که برای هر کدام از جنسیت‌ها باید یک نفر با سن زیر ۳۰ سال و دیگری با سن بیشتر از ۳۰ سال باشد. همچنین صدای ضبط شده نهایتاً باید ۳ ثانیه طول بکشد. در نهایت فایل دیتاست موجود شامل ۱۷۱۵۷ فایل صدا با فرمت wav است.

۲.۲ تمیزسازی دیتاست

در ابتدا برای بررسی دیتاست، فایل CSV داده شده را فراخوانی کرده و به صورت زیر نمایش می‌دهیم:

	voice_id	emotion_id	text_id	gender	age
0	15997	1	1	m	21
1	16001	1	2	m	21
2	16005	1	3	m	21
3	16009	1	4	m	21
4	16013	1	5	m	21
...
16860	10563	4	6	f	54
16861	10567	4	7	f	54
16862	10571	4	8	f	54
16863	10575	4	9	f	54
16864	10539	4	10	f	54

16865 rows × 5 columns

در همین ابتدای کار مشاهده می شود که از ۱۷۱۵۷ فایل صوتی موجود، اطلاعات ۱۶۸۶۴ فایل صوتی داده شده است.

حال بررسی میکنیم که هر کدام از ویژگی های وویس در کل دیتاست (جنسیت، سن، شماره وویس و ...) چند مقدار یکتا و چند مقدار خالی دارند:

	Data Type	Unique Values	Null Values
voice_id	int64	16808	0
emotion_id	int64	10	0
text_id	int64	11	0
gender	object	6	0
age	object	59	0

مشاهده می شود که دیتاست داده شده دارای مشکلات زیر است که باید برطرف شوند :

- تعداد مقادیر یکتای "آیدی احساس" به جای ۴ تا، ۱۰ تا است.

- تعداد مقادیر یکتای "آیدی متن" به جای ۱۰ تا، ۱۱ تا است.

- تعداد مقادیر یکتای "جنسیت" به جای ۲ تا، ۱۰ تا است.

برای درک بیشتر این موضوع هر کدام از ویژگی های مورد نظر را مورد بررسی قرار می دهیم:

emotionID	Unique Values	number of Unique Values
0	1	4191
1	2	4170
2	3	4191
3	4	4193
4	5	20
5	6	20
6	7	20
7	8	20
8	9	20
9	10	20

مشاهده می شود که برای "آیدی احساس"، آیدی های بزرگتر از ۴ به مثابه نویز بوده و باایستی نمونه های با آیدی احساس بزرگتر از ۴ از دیتاست حذف شوند.

textID		
Unique Values	number of Unique Values	
0	0	16
1	1	1723
2	2	1704
3	3	1744
4	4	1727
5	5	1660
6	6	1663
7	7	1659
8	8	1660
9	9	1658
10	10	1651

مشاهده می شود که برای "آیدی متن" ، متن های با آیدی برابر "0" به مثابه نویز بوده و با استیتی نمونه های با آیدی متن برابر "0" از دیتاست حذف شوند.

gender		
Unique Values	number of Unique Values	
0	m	8235
1	f	7585
2	M	503
3	F	382
4	w	80
5	f	80

برای "جنسیت" نیز نمونه هایی که جنسیت با برچسب "w" دارند را از دیتاست حذف می کنیم. همچنین نمونه های با برچسب جنسیت "M" و "m" را برچسب "1" و نمونه های با برچسب جنسیت "f" و "F" را برچسب "0" می دهیم. برای "سن" نیز نمونه های با سن "30-" و "30+" را از دیتاست حذف می کنیم.

حال موارد بالا را به صورت زیر روی دیتاست اعمال می کنیم :

```
df_filtered = df[df['emotion_id'] <= 4]
df_filtered = df_filtered[df_filtered['text_id'] >= 1]
df_filtered = df_filtered.drop(df_filtered[(df_filtered['age'] == '30-') | (df_filtered['age'] == '30+')].index)
df_filtered = df_filtered[df_filtered['gender'] != 'w']

gender = {'m': 1, 'M': 1, 'f': 0, 'F': 0, 'w': 0}

df_filtered.gender = [gender[item] for item in df_filtered.gender]
```

در ادامه تمیزسازی داده، نمونه های تکراری را از دیتاست به صورت زیر حذف می کنیم:

```
# Delete duplicated rows based on voice_id
df_filtered = df_filtered.drop_duplicates(subset=['voice_id'], keep="last")
```

در آخر نیز نمونه هایی که فایل صوتی آنها موجود است ولی مشخصات آن در فایل CSV نیست را حذف می کنیم.

حال دیتاست تمیز شده را بررسی می کنیم:

	Data Type	Unique Values	Null Values
voice_id	int64	16440	0
emotion_id	int64	4	0
text_id	int64	10	0
gender	int64	2	0
age	int64	57	0

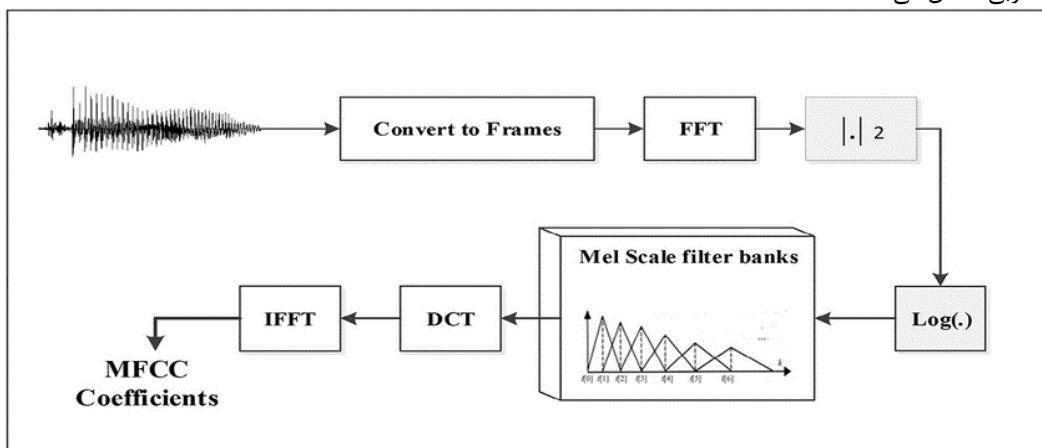
مشاهده می شود که تمامی نواقص گفته شده برطرف شده اند.
با انجام تمیزسازی داده، تعداد نمونه های صوتی در نهایت ۱۶۴۴۰ نمونه است.

۳.۲ استخراج ویژگی

در مرحله اول، باید داده ها را به فرمت قابل فهم برای ماشین بارگذاری کنیم. در ادامه ابتدا و انتهای فایل های صوتی که بدون صدا هستند و به اصطلاح ساینت هستند را از فایل های صوتی جدا می کنیم. سپس مقادیر نمونه های فایل های صوتی را به مقدار ماکریزیم نرمالسازی می کنیم. حال به سراغ کاهش نویز فایل های صوتی می رویم. Noisereduce یک الگوریتم کاهش نویز در پایتون است که نویز را در سیگنال های حوزه زمان مانند گفتار، بیواکوستیک و سیگنال های فیزیولوژیکی کاهش می دهد. حال این الگوریتم را روی تک تک فایل های صوتی اعمال می کنیم.
اکنون به کمک دو روش MFCC و Chroma Frequencies ویژگی استخراج می کنیم.

MFCC (Mel Frequency Cepstral Coefficients) ۱.۳.۲

متداول ترین و کاراترین ویژگیها برای بازناسایی گفتار، ضرایب فرکانسی مل کپستروم (MFCC) هستند. ایده اصلی در استخراج ضرایب خاصیت گوش انسان در دریافت و فهم گفتار است و همین مسئله این ضرایب را به ابزاری قدرتمند در تمامی حوزه هایی پردازش و بازناخت گفتار تبدیل کرده است. تعداد ضرایب مورد استفاده در بازناسایی گفتار معمولاً بین ۱۳ تا ۲۹ است. ضریب صفرم نشان دهنده انرژی است که از آن بعنوان ویژگی شیمر یاد می شود. برای استخراج این ضرایب در مرحله اول، تبدیل فوریه بر سیگنال اعمال می شود. سپس توان طیف به دست آمده به مقیاس مل نگاشته می شود و از توان در هر فرکانس مل، لگاریتم گرفته می شود. در مرحله آخر طیف لگاریتمی مل به حوزه زمان برگردانده می شود. نتیجه این تبدیلات نمایش کپسکال طیف سیگنال گفتار است که ویژگیهای طیفی یک فریم از سیگنال گفتار را به خوبی نشان می دهد:



Chroma Frequencies ۲.۳.۲

ویژگی Chroma Frequencies ارائه‌ای قدرتمند و جالب توجه برای صدای موسیقی است که در آن کل طیف در Bin ۱۲ نشانگر ۱۲ نیم‌گام (semitones) یا کرومای (chroma) از اکتاوهای موسیقی است. از آنجایی که در موسیقی، نت‌هایی که دقیقاً یک اکتاو از هم فاصله دارند، مشابه یکدیگر تلقی می‌شوند، دانستن توزیع کرومای حتی بدون اکتاو اصلی می‌تواند اطلاعات موسیقایی مفیدی در مورد صدا بدهد. تمامی این کارها توسط پکیج *liberosa* پیاده‌سازی شده است.

۴.۲ جمع بندی کارهای انجام شده در این بخش

- چک کردن مقادیر null و unique مشخصات هر وویس (شماره وویس، جنسیت و ...)
- فیکس کردن مقادیر نامعتبر مشخصات هر وویس
- حذف سمپل‌های تکراری
- حذف سمپل‌هایی که مشخصات آن‌ها در فایل CSV نیست
- تناظر یک به یک بین سمپل‌ها و فایل‌ها
- حذف فضای خالی اول و آخر هر وویس
- حذف نویز از وویس
- حذف نمونه‌های صدای با طول بیشتر از ۳ ثانیه
- نرمال کردن مقادیر سمپل وویس
- استخراج و ذخیره فیچرهای mfcc و chroma

فصل ۳

طبقه بندی

۱.۳ مقدمه

در این فصل طبقه بندی داده‌ها را برای احساسات و جنسیت بر روی دو دسته ویژگی Chroma Frequency و MFCC انجام می‌دهیم. برای این منظور از سه طبقه بند KNN، SVM و MLP استفاده می‌کنیم.

۲.۳ طبقه بند (KNN)

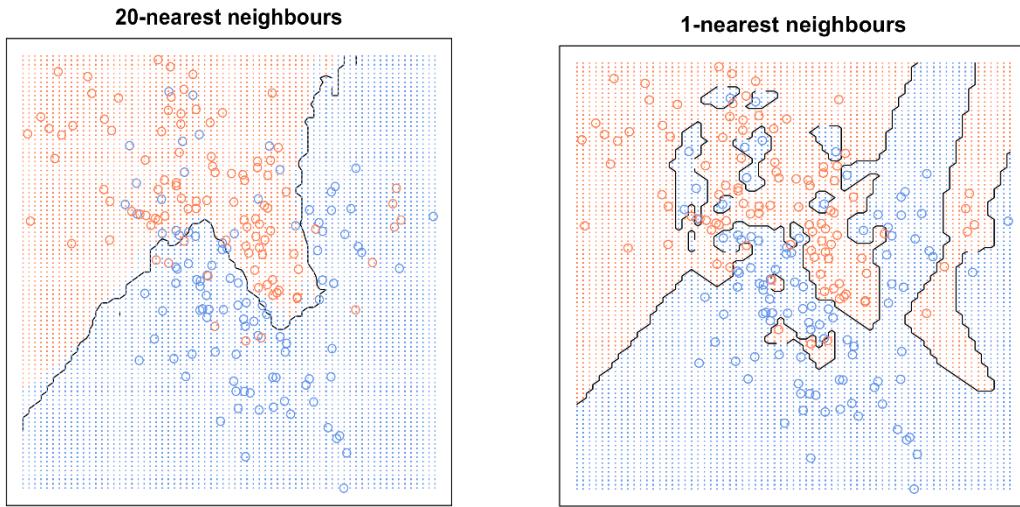
الگوریتم KNN (نزدیکترین همسایه) نوعی از الگوریتم‌های یادگیری ماشین تحت نظرارت است که هم در مسائل طبقه بندی و هم در مسائل رگرسیون پیشگویانه مورد استفاده قرار می‌گیرد. اگرچه، غالباً در مسائل طبقه بندی پیشگویانه، در صنعت از آن استفاده می‌شود. بر اساس آمارهای ارائه شده در وبسایت kdnuggets الگوریتم KNN نزدیکترین همسایگی یکی از ده الگوریتمی است که بیشترین استفاده را در پروژه‌های گوناگون یادگیری ماشین و داده‌کاوی، هم در صنعت و هم در دانشگاه داشته است. همچنین KNN یک الگوریتم یادگیری غیرپارامتری است زیرا درباره داده اصلی هیچ فرضی در نظر نمی‌گیرد.

این الگوریتم دارای دو هایپر پارامتر است :

۱. تعداد همسایه‌ها (K):

اصلی‌ترین hyperparameter این الگوریتم، مقدار k می‌باشد. میدانیم که این الگوریتم از "تشابه ویژگی" برای پیش‌بینی مقادیر نقاط داده جدید استفاده می‌کند، که به این معنی است که k تا از همسایه‌های اطرافش را در نظر می‌گیرد و بر اساس آن‌ها رأی‌گیری درباره وضعیت تعلق یک نمونه داده به کلاس‌های موجود انجام می‌شود.

تأثیر مقدار K بر مراز تصمیم گیری:



مقدار k کوچک باعث پیچیده‌تر شدن مرز کلاس‌ها روان‌تر می‌شود. با افزایش مقدار k مرز کلاس‌ها روان‌تر می‌شود. همچنین اگر k را خیلی کوچک در نظر بگیریم، الگوریتم KNN به نقاط نویزی حساس می‌شود.

۲. متريک شباهت (فاصله):

يک ديجر از مواردي اين الگوريتم نقش تاثير گذاري ايفا كند، distance metric مي‌باشد و به اندازه قابل توجهی می‌تواند روی دقت مدل خروجي تاثير گذار باشد. چرا كه در واقع اين عامل است كه نزديكی همسایه‌ها را تعريف می‌کند. بطوري كه ممکن نقطه‌اي خاص و ثابت در يك متريک جزو نزديك ترين همسایه‌ها به شمار بيايد و در متريک ديجر اينگونه نباشد. برای محاسبه فاصله در KNN معمولاً از ۳ متريک زير استفاده می‌شود:

Distance functions

Euclidean

$$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

Manhattan

$$\sum_{i=1}^k |x_i - y_i|$$

Minkowski

$$\left(\sum_{i=1}^k (|x_i - y_i|)^q \right)^{1/q}$$

از ويزگي‌های مثبت اين طبقه‌بند كه باعث انتخاب آن در اين پروژه شده است عبارت است از :

- پياده سازي ساده

- صفر بودن هزينه مرحله آموزش

- الگوریتمی بسیار ساده برای فهم و تفسیر است
- از آنجایی که در این الگوریتم هیچ فرضی درباره داده نمی شود، برای داده های غیر خطی بسیار مناسب است

تعیین مقادیر مناسب هایپر پارامترها

برای این کار از متدهای Halving Grid search استفاده می کنیم و برای دو هایپر پارامتر موردنظر مقادیر زیر را در نظر میگیریم:

- تعداد همسایه (K) : ۳ ، ۵ ، ۷ ، ۹ ، ۱۱ ، ۱۵ ، ۲۰ ، ۲۵ ، ۳۰ ، ۴۰ ، ۵۰

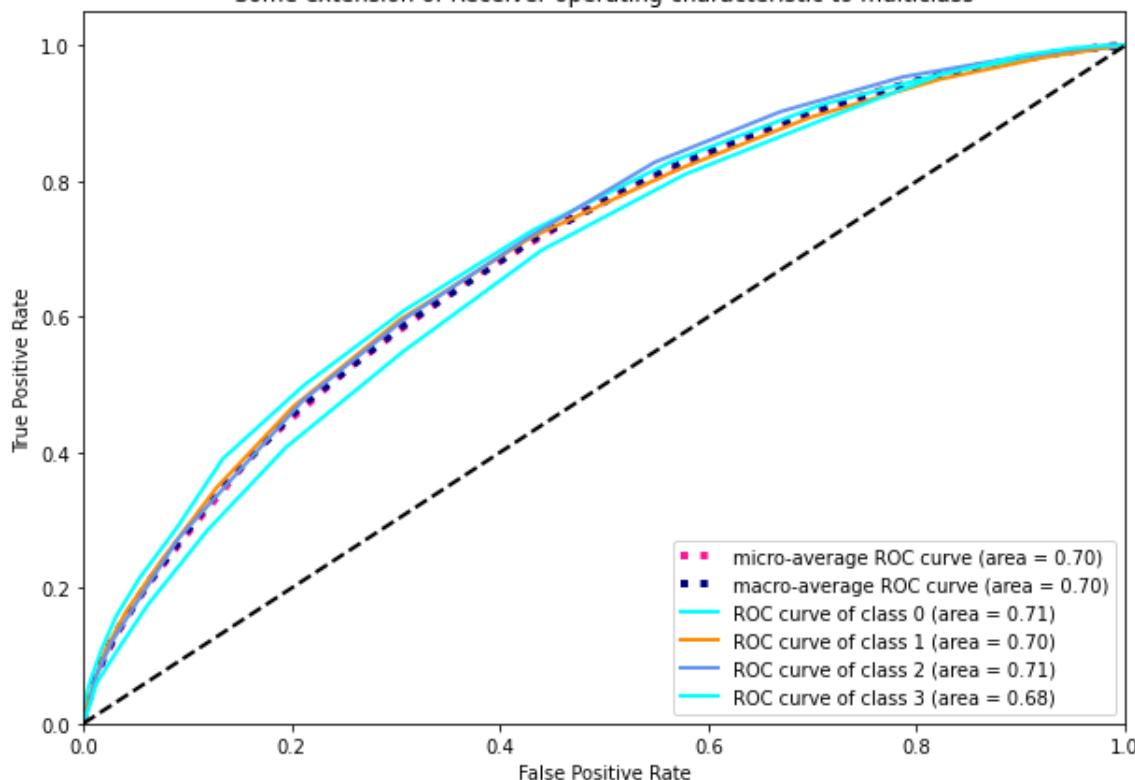
- متریک: اقلیدسی، منهتن، مینکوفسکی

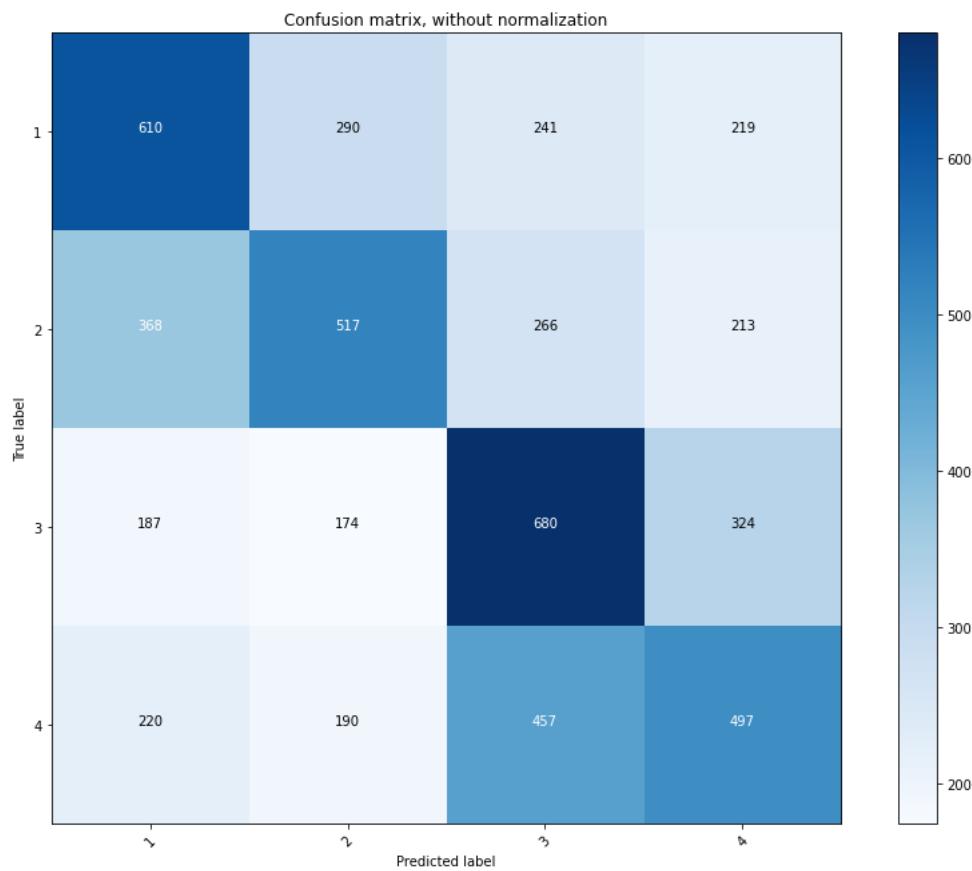
بهترین نتیجه بدست آمده برای متریک "منهتن" با تعداد همسایه برابر ۲۵ است. حال الگوریتم KNN را با این مقادیر روی داده ها اعمال می کنیم.

۱.۲.۳ اعمال KNN برای طبقه بندی "احساس" با استفاده از دسته ویژگی MFCC :

با استفاده از این روش به دقت کلی ۰.۴۲ رسیده ایم. دیگر نتایج بدست آمده به صورت زیر است:

Some extension of Receiver operating characteristic to multiclass





	classes	Recall	Precision	F1-Score
0	Angry	0.44852941	0.44043321	0.44444444
1	Sad	0.37903226	0.44150299	0.40788955
2	Happy	0.4981685	0.4136253	0.4519774
3	Neutral	0.3643695	0.39664804	0.37982423

تحلیل نتایج:

- کمترین دقت طبقه بندی مربوط به کلاس خنثی و بیشترین دقت طبقه بندی به ترتیب برای کلاس خوشحالی و عصبانیت بوده است.
- حدود ۳۰ درصد از کل نمونه‌ها به کلاس خوشحالی نسبت داده شده اند یعنی کلاسیفایر KNN در اینجا کلاس خوشحالی را با احتمال بیشتری نسبت به بقیه کلاس‌ها طبقه بندی می‌کند.
- حدود ۲۱ درصد از کل نمونه‌ها به کلاس ناراحتی نسبت داده شده اند یعنی کلاسیفایر KNN در اینجا کلاس ناراحتی را با احتمال کمتری نسبت به بقیه کلاس‌ها طبقه بندی می‌کند.

۲.۲.۳ اعمال KNN برای طبقه بندی "احساس" با استفاده از دسته ویژگی Chroma Frequency :

تعیین مقادیر مناسب هایپر پارامترها:

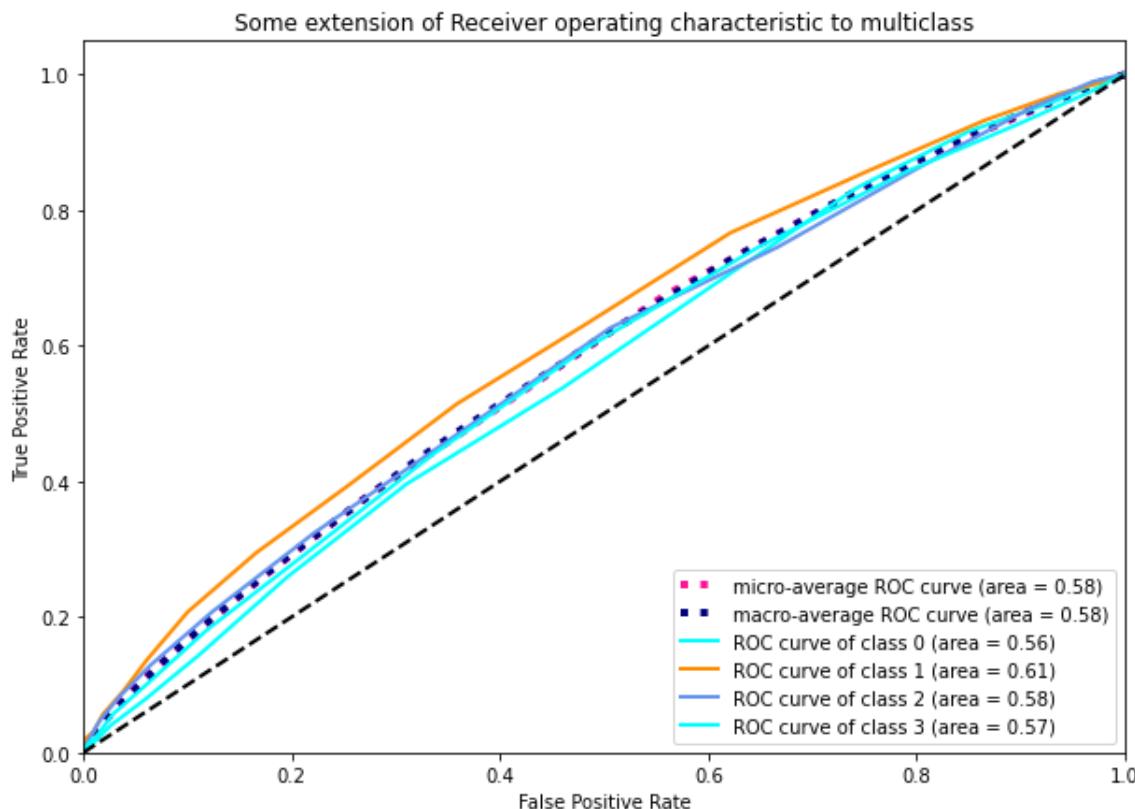
برای این کار از متدهای Halving Grid search استفاده می‌کنیم و برای دو هایپر پارامتر موردنظر مقادیر زیر را در نظر می‌گیریم:

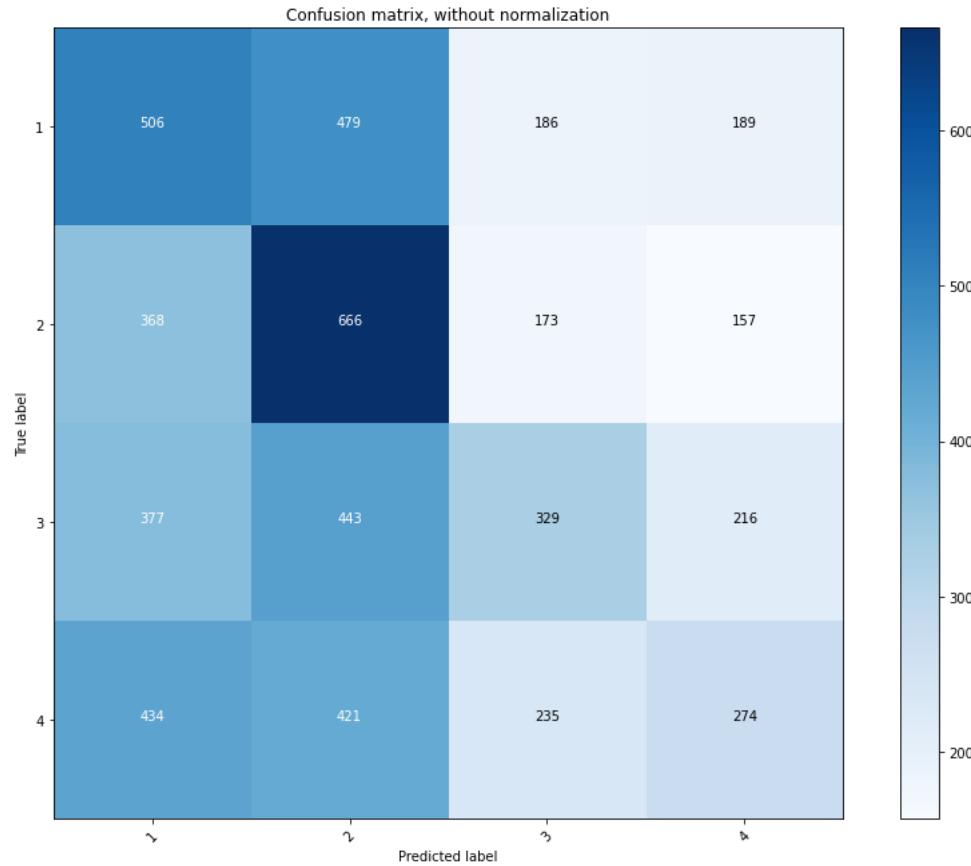
- تعداد همسایه (k) : ۳، ۵، ۷، ۹، ۱۱، ۱۵، ۲۰، ۲۵، ۳۰، ۴۰، ۵۰

- متریک: اقلیدسی، منهتن، مینکوفسکی

بهترین نتیجه بدست آمده برای متریک "منهتن" با تعداد همسایه برابر ۱۱ است. حال الگوریتم KNN را با این مقادیر روی داده‌ها اعمال می‌کنیم

با استفاده از این روش به دقت کلی ۰.۳۳ رسیده‌ایم. دیگر نتایج بدست آمده به صورت زیر است:





classes		Recall	Precision	F1-Score
0	Angry	0.37205882	0.30029674	0.33234811
1	Sad	0.48826979	0.33150821	0.39490068
2	Happy	0.24102564	0.35644637	0.28758741
3	Neutral	0.20087977	0.3277512	0.24909091

تحلیل نتایج:

- کلاس ناراحتی با دقت بیشتری نسبت به بقیه طبقه‌بندی شده‌اند و کلاس خنثی کمترین دقت طبقه‌بندی را داشته است.
- ۳۶.۸ درصد از کل نمونه‌ها به کلاس ناراحتی نسبت داده شده اند یعنی کلاسیفایر KNN در اینجا کلاس ناراحتی را با احتمال بیشتری نسبت به بقیه کلاس‌ها طبقه‌بندی می‌کند.
- ۱۵.۳ درصد از کل نمونه‌ها به کلاس خنثی نسبت داده شده اند یعنی کلاسیفایر KNN در اینجا کلاس خنثی را با احتمال کمتری نسبت به بقیه کلاس‌ها طبقه‌بندی می‌کند.
- با مقایسه نتایج KNN روی دسته ویژگی‌های Chroma و MFCC می‌توان گفت که در مجموع کلاس عصبانیت با دقت بیشتری نسبت به بقیه کلاس‌ها طبقه‌بندی شده‌اند که نتیجه‌ای کاملاً قابل پیش‌بینی است زیرا تشخیص احساس عصبانیت توسط انسان نیز نسبت به

بقيه احساسات با دقت بيشرى انجام می‌گيرد.

- با مقایسه نتایج KNN روی دسته ویژگی‌های Chroma و MFCC می‌توان گفت که در مجموع کلاس خنثی با خطای بيشرى تخمين زده می‌شود که بازهم نتیجه بدست آمده در واقعیت توسط انسان هم قابل توجيه است.
- با مقایسه نتایج KNN روی دسته ویژگی‌های Chroma و MFCC می‌توان گفت نتایج طبقه بندی بر روی MFCC بهوضوح بهتر از Chroma است درواقع قابلیت تفکیک پذیری MFCC بيشتر بوده است و در نتیجه دقت بيشرى داشته است از آنجایی که MFCC در حوزه فرکانس، ویژگی استخراج می‌کند می‌توان گفت که داده‌ها در حوزه فرکانس تفکیک‌پذیر از حوزه زمان هستند.
- در نهايیت می‌توان گفت با توجه به اينکه داده‌های مورد نظر تفکیک پذیری بالايی ندارند و حتی توسط انسان هم با خطای طبقه بندی می‌شوند در نتیجه نمی‌توان انتظار دقت بالاي طبقه بندی را داشت.

۳.۳ طبقه بند MLP

پرسپترون‌های چندلایه یا به اختصار MLP، نوع کلاسیک شبکه‌های عصبی میباشد MLP ها بسیار انعطاف پذیر می‌باشند و می‌توان از آنها برای یادگیری و mapping از ورودی به خروجی که همان لیبل‌ها می‌باشند استفاده کرد و این انعطاف پذیری به آنها اجازه می‌دهد تا برای مسائل طبقه بندی دو، یا چندکلاسه با انواع داده‌ها مثل تصویر، متن و... مناسب باشند. در نتیجه ما MLP را به عنوان دومین الگوریتم طبقه بندی برای مسئله خود استفاده می‌کنیم. در ادامه قصد داریم با ۴ تا از Hyperparameter های این الگوریتم آشنا شویم:

۱. Solver

این پارامتر در واقع اشاره به نوع الگوریتم optimization ای دارد که باید بکار گرفته شود. طبق اسناد موجود در scikit-learn.org سه مقدار برای این پارامتر وجود دارد که در ادامه می‌توانید مشاهده کنید:

- Lbfgs: این الگوریتم یک نوع optimizer از خانواده متدهای quasi-Newton می‌باشد.
- Sgd: در واقع همان الگوریتم stochastic gradient descent می‌باشد.
- Adam: این الگوریتم به یک optimizer مبتنی بر گرادی ان تصادفی اشاره دارد که توسط Jimmy Diederik Kingma و تواند برای این پارامتر وجود دارد که در ادامه می‌توانید مشاهده کنید:

Adam عموماً رو مجموعه داده‌های بزرگ (هزاران نمونه آموزشی و بيشرى) کاربرد دارد و از نظر هر دو معیار training time و validation score نتیجه بهتری را نسبت به سایر solver ها ارائه می‌دهد. از طرفی lbfgs می‌تواند برای دیتاست‌های کوچک مناسب‌تر باشد و سریعتر همگرا می‌شود.

۲. Learning Rate

نرخ یادگیری یک هايپرپارامتر برای تنظیم کردن الگوریتم های بهینه سازی می‌باشد. این امر با تعیین سایز گام‌ها در جهتی است که تابع کمترین افت را در هر گام دارد. نکته ای که باید حتماً به آن توجه کرد آن است که این پارامتر تنها زمانی کاربرد دارد که از solver های Adam و sdg استفاده کنیم.

Alpha .۳

آلfa پارامتری برای Regularization است که penalty نیز نامیده می شود و با محدود کردن اندازه weight ها با overfitting مبارزه می کند. افزایش آلfa می تواند واریانس بالا (نشانهای از overfitting) را برطرف کند، که در نتیجه آن یک انحنای کمتر خواهیم داشت.

Activation Function .۴

این پارامتر در واقع Activation function یا همانتابع فعال ساز hidden layer را مشخص می کند و طبق اسناد موجود در scikit-learn.org چهار مقدار برای این پارامتر وجود دارد که در ادامه می توانید مشاهده کنید:

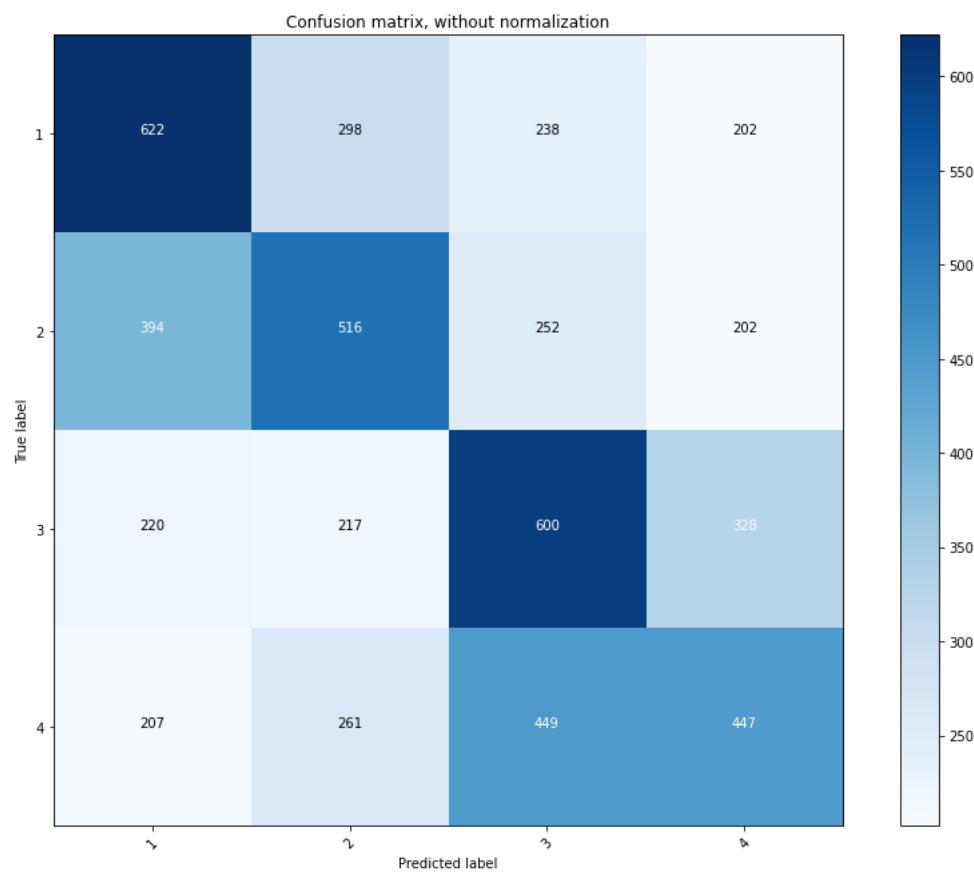
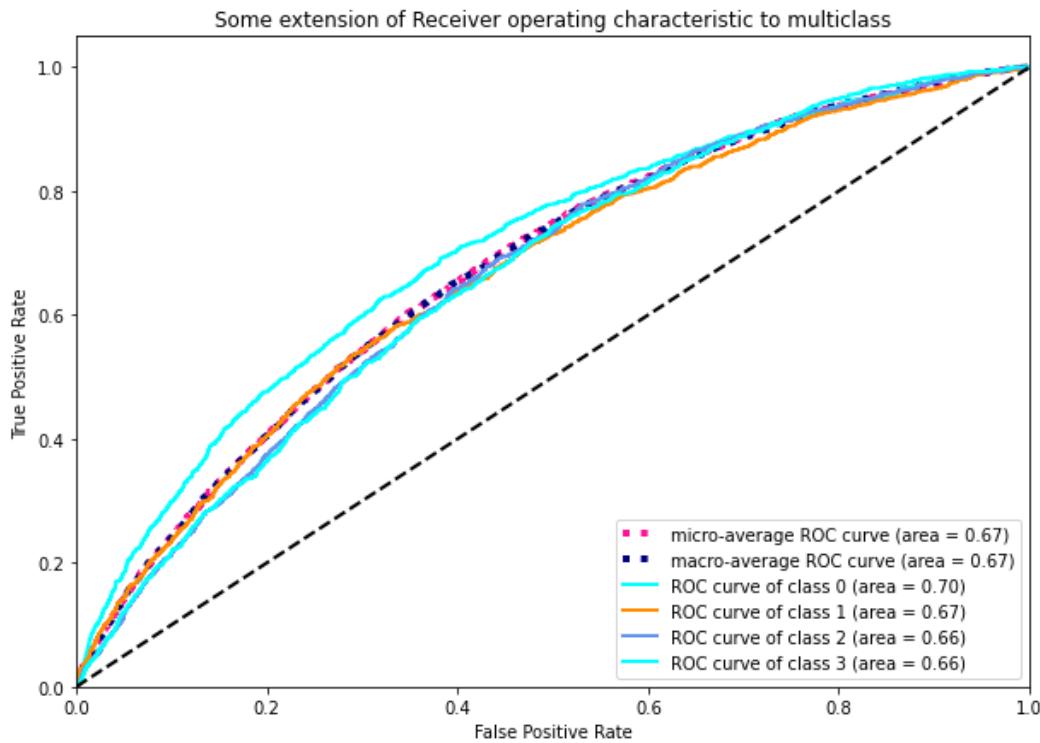
- Identity
- Logistic
- Tanh
- Relu

۱.۳.۳ اعمال MLP برای طبقه بندی "احساس" با استفاده از دسته ویژگی : MFCC

در ابتدا با استفاده از متod Halving Grid search مقادیر هایپرپارامترها را تعیین می کنیم. بهترین پارامترهای بدست آمده به شرح زیر است:

- Solver : sgd
- Learning rate : Adaptive
- Alpha : 0.05
- Activation : tanh

با استفاده از این روش به دقت کلی ۴۰٪ رسیده ایم. دیگر نتایج بدست آمده به صورت زیر است :



	classes	Recall	Precision	F1-Score
۰	Angry	0.46	0.43	0.44
۱	Sad	0.38	0.4	0.39
۲	Happy	0.44	0.39	0.41
۳	Neutral	0.33	0.38	0.35

تحلیل نتایج:

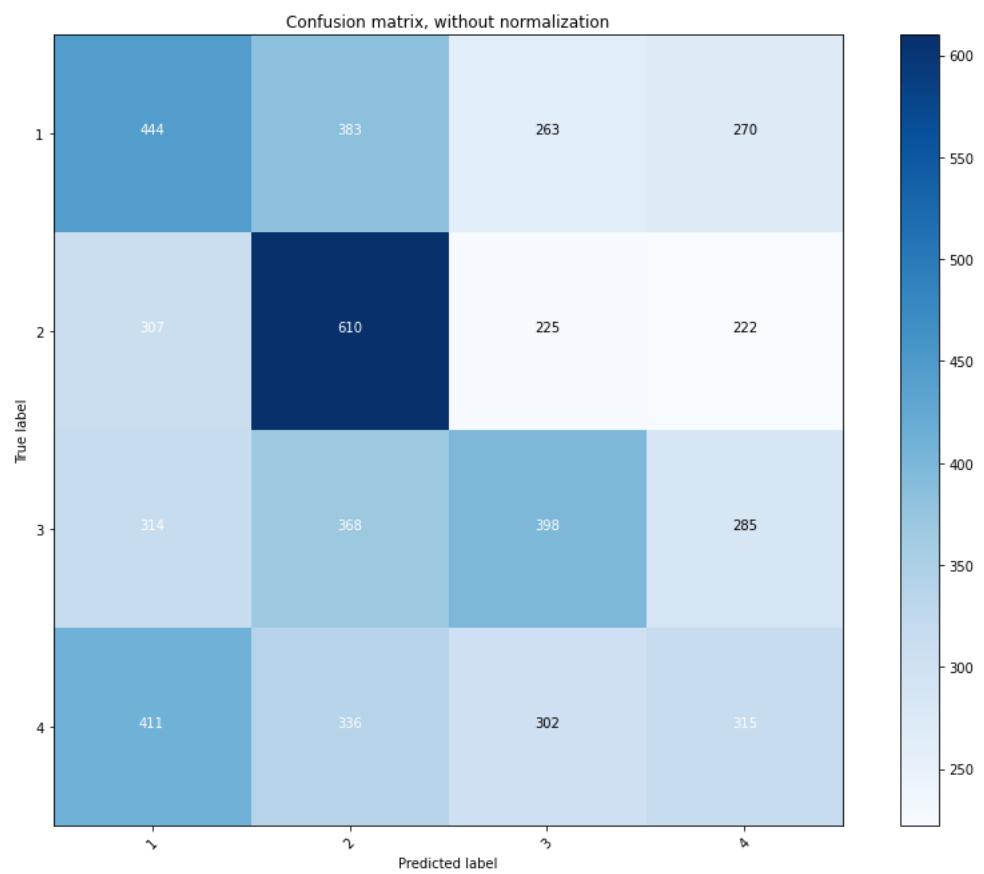
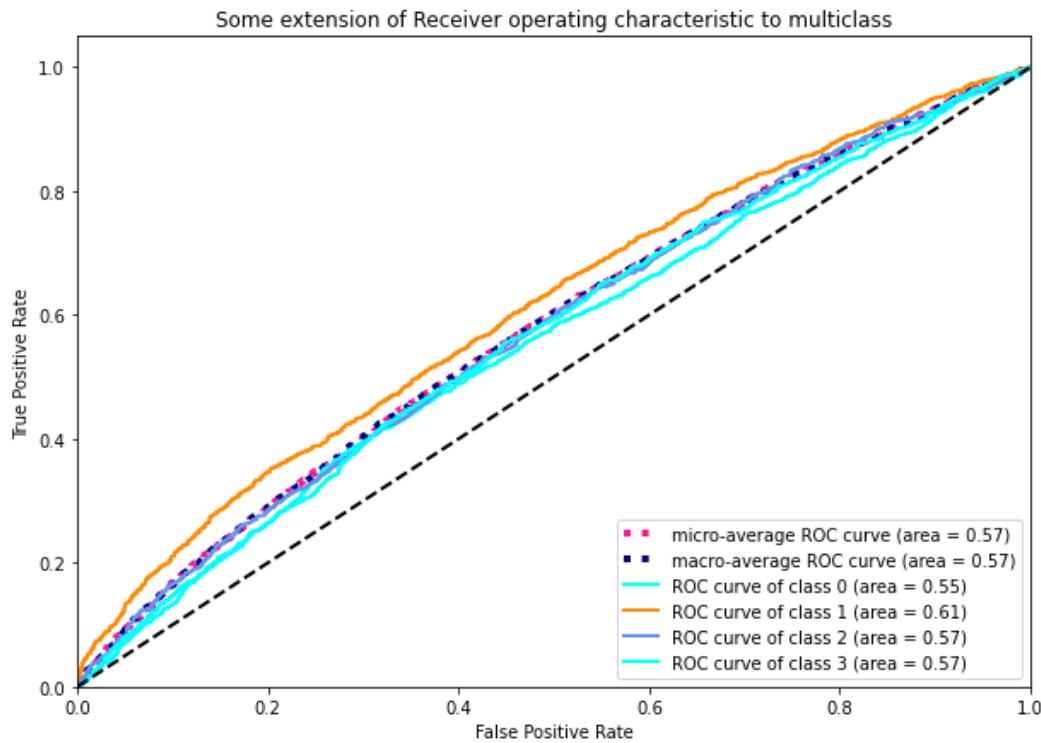
- کلاس عصبانیت با دقت بیشتری نسبت به بقیه طبقه بندی شده‌اند و کلاس خنثی کمترین دقت طبقه بندی را داشته است.
- ۲۸ درصد از کل نمونه‌ها به کلاس خوشحالی نسبت داده شده اند یعنی کلاسیفایر MLP در اینجا کلاس خوشحالی را با احتمال بیشتری نسبت به بقیه کلاس‌ها طبقه بندی می‌کند.
- ۲۱ درصد از کل نمونه‌ها به کلاس خنثی نسبت داده شده اند یعنی کلاسیفایر MLP در اینجا کلاس خنثی را با احتمال کمتری نسبت به بقیه کلاس‌ها طبقه بندی می‌کند.
- بیشترین خطا مربوط به کلاس خنثی است که بیشتر از بقیه با کلاس خوشحالی اشتباه گرفته شده است.
- دقت کلی MLP بر روی دسته ویژگی MFCC نسبت به KNN دو درصد کمتر است.
- با مقایسه نتایج MLP و KNN می‌توان گفت که کلاس عصبانیت درمجموع با دقت بهتری نسبت به بقیه طبقه‌بندی می‌شود که نتیجه بدست آمده منطقی می‌باشد و نیروی انسانی هم احتمالاً کلاس عصبانیت را بهتر از بقیه کلاس‌ها بتواند دسته‌بندی کند. همچنین می‌توان گفت که در مجموع کلاس خنثی با خطای بیشتری تخمین زده می‌شود که بازهم نتیجه بدست آمده در واقعیت توسط انسان هم قابل توجیه است.

۲.۳.۳ اعمال MLP برای طبقه بندی "احساس" با استفاده از دسته ویژگی Chroma Frequency

در ابتدا با استفاده از متد Halving Grid search مقادیر هایپرپارامترها را تعیین می‌کنیم. بهترین پارامترهای بدست آمده به شرح زیر است :

- Solver : sgd
- Learning rate : Adaptive
- Alpha : 0.01
- Activation : Relu

با استفاده از این روش به دقت کلی ۳۲٪ رسیده‌ایم. دیگر نتایج بدست آمده به صورت زیر است:



classes		Recall	Precision	F1-Score
0	Angry	0.33	0.3	0.31
1	Sad	0.45	0.36	0.4
2	Happy	0.29	0.34	0.31
3	Neutral	0.23	0.29	0.26

تحلیل نتایج:

- برخلاف نتایج قبلی که کلاس عصبانیت با دقت بیشتری نسبت به بقیه طبقه بندی شده بود در اینجا کلاس ناراحتی با دقت بیشتر طبقه بندی شده است همچنین مشابه نتایج بدست آمده در قبل، کلاس خنثی کمترین دقت طبقه بندی را داشته است.
- ۳۱ درصد از کل نمونه‌ها به کلاس ناراحتی نسبت داده شده اند یعنی کلاسیفایر MLP در اینجا کلاس خوشحالی را با احتمال بیشتری نسبت به بقیه کلاس‌ها طبقه بندی می‌کند.
- ۲۰ درصد از کل نمونه‌ها به کلاس خنثی نسبت داده شده اند یعنی کلاسیفایر MLP در قبل بدست آمد نیز کلاس خنثی را با احتمال کمتری نسبت به بقیه کلاس‌ها طبقه بندی می‌کند.
- بیشترین خطا مربوط به کلاس خنثی است که حتی از حالت شناسی نیز دقت پایین‌تری داشته است همچنین این کلاس بیشتر از بقیه با کلاس عصبانیت اشتباہ گرفته شده است.
- دقت کلی MLP بر روی دسته ویژگی Chroma نسبت به KNN یک درصد کمتر است یعنی در کل تفاوت چندانی با یکدیگر نداشته‌اند.
- با مقایسه نتایج MLP روی دسته ویژگی‌های Chroma و MFCC می‌توان گفت که دسته ویژگی‌های MFCC قابلیت تفکیک‌پذیری بیشتری نسبت به دسته ویژگی Chroma داشته و در نتیجه دقت بالاتری خواهد داشت.
- با مقایسه نتایج MLP و KNN روی دسته ویژگی‌های Chroma و MFCC می‌توان گفت کلاس خنثی قابلیت تفکیک‌پذیری پایین‌تری نسبت به بقیه کلاس‌ها دارد و درنتیجه با خطای بیشتری نسبت به بقیه کلاس‌ها تخمین زده می‌شود.

۴.۳ طبقه بند SVM

الگوریتم‌های زیادی برای طبقه بندی در یادگیری ماشین استفاده می‌شوند، اما SVM بهتر از بسیاری از الگوریتم‌های دیگر مورد استفاده است چراکه تجربه نشان داده است نسبت به سایر الگوریتم‌های طبقه بندی دقت بهتری در نتایج دارد. این الگوریتم یکی از پرکاربرد ترین روش‌ها برای طبقه بندی‌های چندکلاسه نیز می‌باشد. البته در حالت پایه ای، SVM از طبقه بندی چند کلاسه پشتیبانی نمی‌کند بدین منظور، از اصل تجزیه مسئله به مسائل طبقه‌بندی فرعی و کوچکتر استفاده می‌شود که همه آنها مسائل طبقه‌بندی بازنری می‌باشند. در SVM هدف این است که خط جداکننده به گونه‌ای انتخاب شود که حاشیه بین دو کلاس تاحد امکان ماکزیمم شود. در ادامه با هایپرپارامترهای SVM آشنا می‌شویم:

۱. Regularization

این پارامتر در کتابخانه skLearn به عنوان پارامتر C می‌شناسیم و به نوعی می‌توان گفت که بیانگیر آن است که تا چه حد می‌خواهیم از misclassification برای داده‌های train جلوگیری کنیم. یک مقدار بزرگ برای این پارامتر موجب می‌شود مدل کم ترین خطای

دسته بندی را برای نمونه های train داشته باشد. مسلماً این صورت احتمال وقوع خطأ برای داده های test بالا می رود و به عبارت دیگر با overfitting مواجه خواهیم شد. در صورتی که مقدار کوچکی را برای این پارامتر تعیین کنیم باعث تولید یک مدل خیلی ساده و به نوعی موجب از دست رفتن رابطه بین feature vector ها و خروجی هدف میشود و در نهایت چیزی که نسیب ما میشود خطأ های زیاد در داده های test و همینطور train است. در واقع اینجا با مسئله معروف bias-variance tradeoff مواجه هستیم و برای رسیدن به بهترین خروجی لازم است این پارامتر با مناسب ترین عدد مقدار دهی شود.

۲. Gamma

پارامتر گاما تعیین می کند که تاثیر هر training sample تا چه حد باشد، که مقادیر کم آن به معنی تاثیر بیشتر و مقادیر زیاد آن به معنی تاثیر کمتر است. به عبارت دیگر، با گاما کم، نقاط دور از separating hyperplane یا همان مرز تصمیم احتمالی برای محاسبه hyperplane در نظر گرفته می شوند. و با گاما بالاتر، نقاط نزدیک hyperplane احتمالی در محاسبه در نظر گرفته می شوند.

۳. Kernel

الگوریتم های SVM از مجموعه ای از توابع ریاضی که به عنوان کرنل تعریف می شوند، استفاده می کنند. وظیفه کرنل این است که داده ها را به عنوان ورودی گرفته و آن ها را به شکل مورد نیاز تبدیل کند. الگوریتم های مختلف SVM، از انواع مختلف توابع کرنل استفاده می کنند. این توابع می توانند انواع متفاوتی داشته باشند. به عنوان مثال خطی، غیرخطی، چند جمله ای، تابع پایه شعاعی (RBF) و سیگموئید. در ادامه بخش زیر سه تا از پر کاربرد ترین kernel ها را معرفی می کنیم:

- کرنل RBF:

این kernel یکی از ترجیح داده شده ترین و مورد استفاده ترین توابع در SVM است. معمولاً برای داده های غیر خطی انتخاب می شود و زمانی استفاده می شود که اطلاعات قبلی در مورد داده ها وجود نداشته باشد.

- کرنل Polynomial:

این کرنل را می توان به نوعی یک بازنمایی عمومی تر از کرنل خطی دانست البته به اندازه کرنل های دیگر ترجیح داده نمی شود چراکه عموماً نسبت به آن ها دقت کمتری دارد اما در زمینه پردازش تصویر بسیار محبوب و پر کاربرد است.

- کرنل Linear:

این تابع از پایه ای ترین نوع کرنل ها می باشد و عموماً سریعتر از کرنل های دیگر می باشد. تجربه نشان داده است که این کرنل زمانی که تعداد فیچرها زیاد است بهترین عملکرد را ارائه می دهد. این کرنل در زمینه پردازش متن بسیار محبوب و پر کاربرد است.

از ویژگی های مثبت این طبقه بند که باعث انتخاب آن در این پروژه شده است عبارت است از:

- وقتی فاصله خوبی بین کلاس های مختلف وجود داشته باشد، نسبتاً خوب کار می کند.
- در فضاهای با ابعاد بالاتر کارایی بیشتری دارد.
- در شرایطی که تعداد ابعاد بیش از تعداد نمونه ها باشد نیز کار می کند.
- یک زیر مجموعه از نقاط تمرینی را در تابع تصمیم گیری استفاده می کند (که به آن ها بردارهای پشتیبان گفته می شود)، بنابراین در مصرف حافظه نیز به صورت بهینه عمل می کند.
- برای داده هایی که هیچ ایده ای در مورد آنها نداریم بسیار خوب عمل می کند.

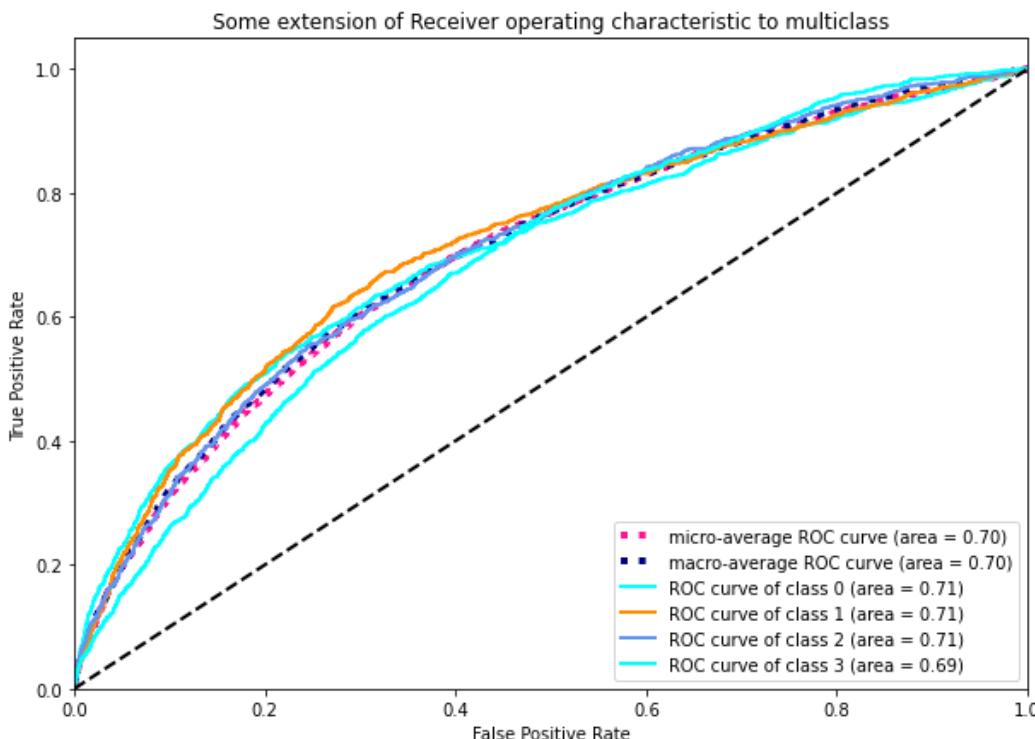
- با یکتابع کرنل مناسب تقریبا هر مسئله پیچیده ای را می تواند حل کند.
- برخلاف شبکه های عصبی فقط به صورت محلی بهینه نیست.
- برای داده های با ابعاد بالا خوب عمل می کند.
- خطر over-fitting در SVM کم است.

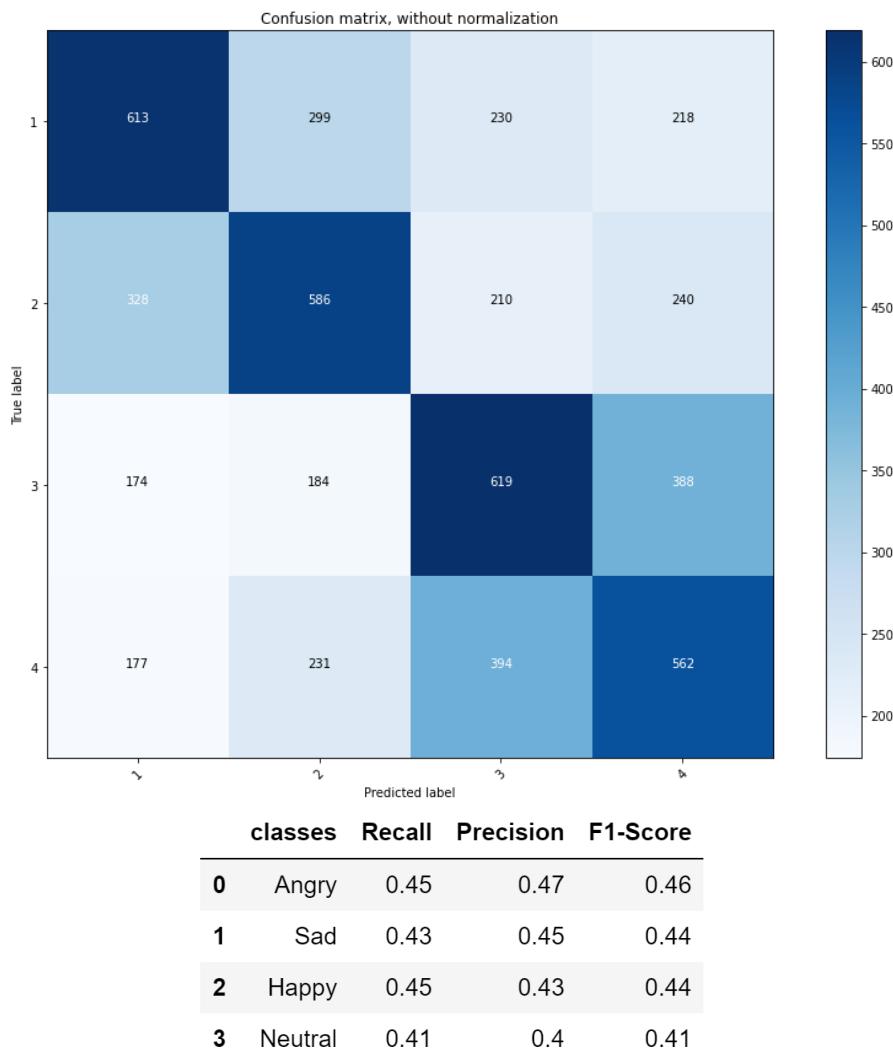
۱.۴.۳ اعمال SVM برای طبقه بندی "احساس" با استفاده از دسته ویژگی MFCC:

در ابتدا با استفاده از متدهای Halving Grid search مقادیر هایپر پارامترها را تعیین می کنیم. بهترین پارامترهای بدست آمده به شرح زیر است:

- Kernel : RBF
- Regularization : 0.5
- Gamma : 0.1

با استفاده از این روش به دقت کلی ۰.۴۴ رسیده ایم. دیگر نتایج بدست آمده به صورت زیر است:





تحلیل نتایج:

- طبقه بند SVM دقیق بالاتری نسبت به دو طبقه بند قبلی روی دسته ویژگی های MFCC داشته است.
- تمامی کلاس ها در این طبقه بند با دقیق نزدیک به هم دسته بندی شده اند و سطح زیر نمودار ROC هریک بیانگر این موضوع نیز می باشد چرا که سه کلاس مساحت یکسان و کلاس خنثی مساحت نزدیک به این سه کلاس داشته است.
- در اینجا نیز کلاس عصبانیت در کنار کلاس خوشحالی بیشترین دقیق را داشته است و همانند کلاسیفایرهای KNN و MLP برای دسته ویژگی های MFCC کلاس خنثی با بیشترین خطأ همراه بوده است.

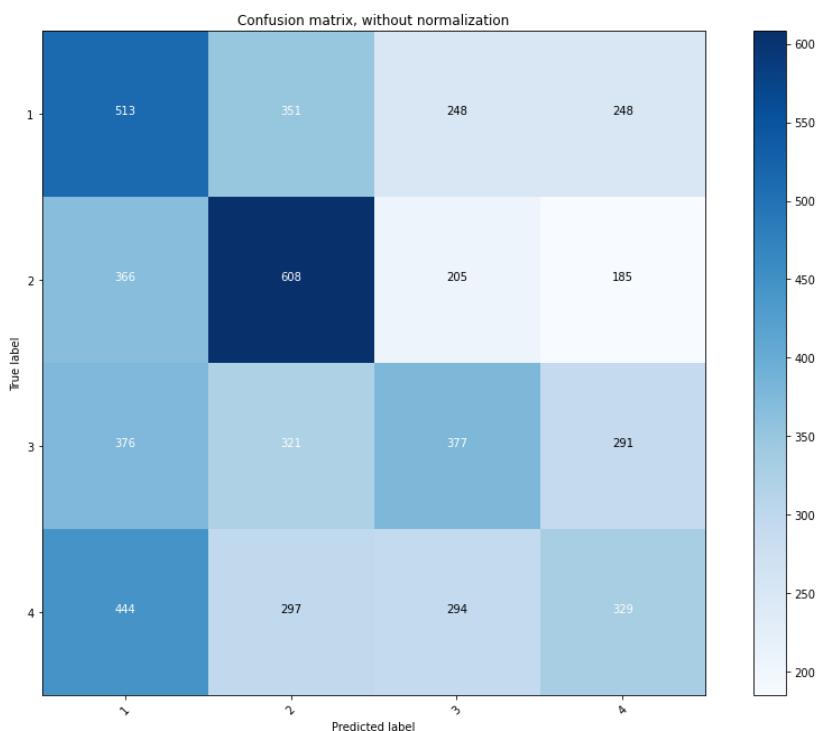
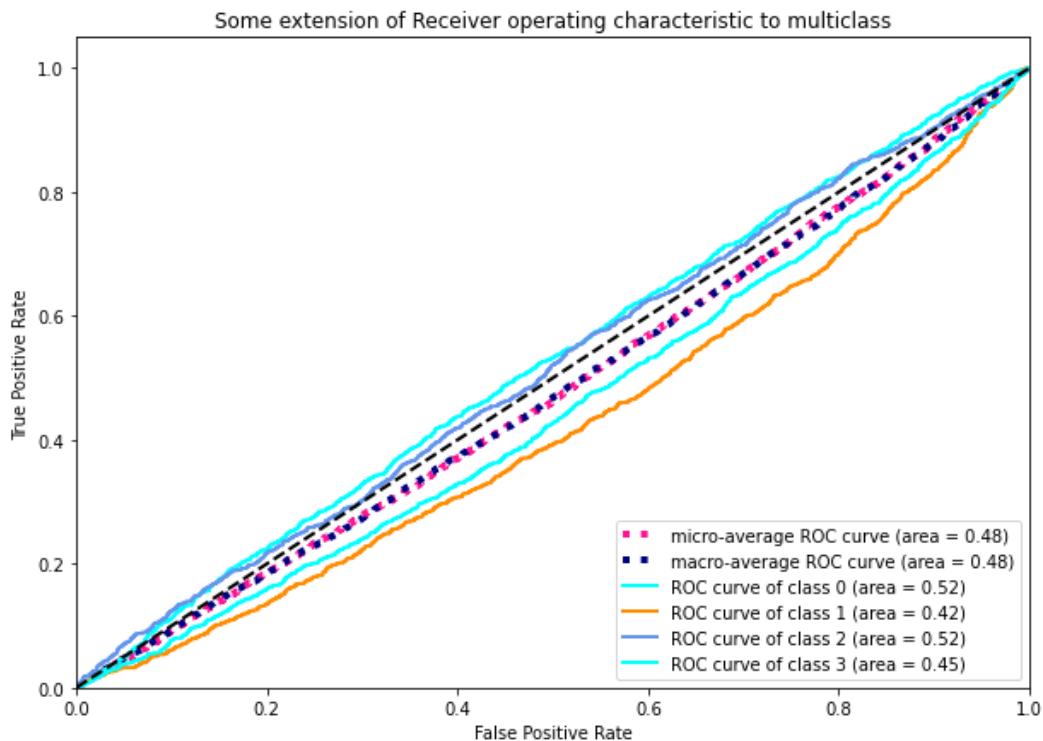
۲.۴.۳ اعمال SVM برای طبقه بندی "احساس" با استفاده از دسته ویژگی Chroma Frequency

در ابتدا با استفاده از متدهای Halving Grid search مقادیر هایپر پارامترها را تعیین می کنیم. بهترین پارامترهای بدست آمده به شرح زیر است:

- Kernel : RBF

- Regularization : 0.5
- Gamma : 0.1

با استفاده از این روش به دقت کلی ۰.۳۴ رسیده ایم. دیگر نتایج بدست آمده به صورت زیر است:



	classes	Recall	Precision	F1-Score
0	Angry	0.377	0.301	0.335
1	Sad	0.445	0.385	0.413
2	Happy	0.276	0.335	0.302
3	Neutral	0.241	0.312	0.272

تحلیل نتایج:

- طبقه بند SVM دقیق بالاتری نسبت به دو طبقه بند قبلی روی دسته ویژگی های Chroma داشته است.
- کلاس ناراحتی بیشترین دقیق و کلاس خنثی بیشترین خطای را داشته است.
- ۱۹ درصد از کل نمونه ها به کلاس خنثی نسبت داده شده اند یعنی کلاسیفایر SVM در اینجا مشابه نتایجی که در قبل بدست آمد نیز کلاس خنثی را با احتمال کمتری نسبت به بقیه کلاس ها طبقه بندی می کند.

جمع بندی نهایی نتایج طبقه بندی احساس با استفاده از دسته ویژگی های Chroma و MFCC

خلاصه نتایج بدست آمده توسط کلاس بندی SVM و MLP و KNN بر روی دو دسته ویژگی Chroma Frequency و MFCC به

صورت زیر است:

Chroma Frequency					MFCC						
	classes	KNN	MLP	SVM	Average		classes	KNN	MLP	SVM	Average
0	Accuracy of Angry	0.37	0.33	0.37	0.35	0	Accuracy of Angry	0.45	0.46	0.45	0.46
1	Accuracy of Sad	0.49	0.45	0.44	0.46	1	Accuracy of Sad	0.38	0.38	0.43	0.40
2	Accuracy of Happy	0.24	0.29	0.26	0.26	2	Accuracy of Happy	0.48	0.44	0.45	0.46
3	Accuracy of Neutral	0.2	0.23	0.24	0.22	3	Accuracy of Neutral	0.36	0.33	0.41	0.36
4	totala Accuracy	0.33	0.32	0.34	0.33	4	totala Accuracy	0.42	0.40	0.44	0.42

تحلیل نتایج:

- همان طور که مشاهده می شود دقیق کلی هر یک از طبقه بندها روی دسته ویژگی های Chroma بیشتر از دسته ویژگی های MFCC بوده است به طوری که به طور متوسط دقیق روی MFCC حدود ۹ درصد بالاتر از دقیق روی Chroma بوده است.
- دو کلاس عصبانیت و خوشحالی توسط طبقه بندی روی دسته ویژگی های MFCC با دقیق بالاتری نسبت به بقیه کلاس ها دسته بندی شده اند و کلاس ناراحتی توسط طبقه بندی روی دسته ویژگی های Chroma دقیق بالاتری نسبت به سایر کلاس ها داشته است.
- دقیق کلاس خنثی توسط طبقه بندی روی هر دو دسته ویژگی از بقیه کلاس پایین تر بوده است و می توان گفت که این کلاس بیشتر از بقیه با دیگر کلاس ها اشتباہ گرفته می شود و درنتیجه تفکیک پذیری کمتری نسبت به بقیه داشته است.
- از آنجایی که نتایج طبقه بندی بر روی MFCC به وضوح بهتر از Chroma است در واقع قابلیت تفکیک پذیری MFCC نسبت به Chroma بیشتر بوده است و در نتیجه دقیق بیشتری داشته است از آنجایی که MFCC در حوزه فرکانس، ویژگی استخراج می کند می توان گفت که داده ها در حوزه فرکانس تفکیک پذیری بیشتری دارند

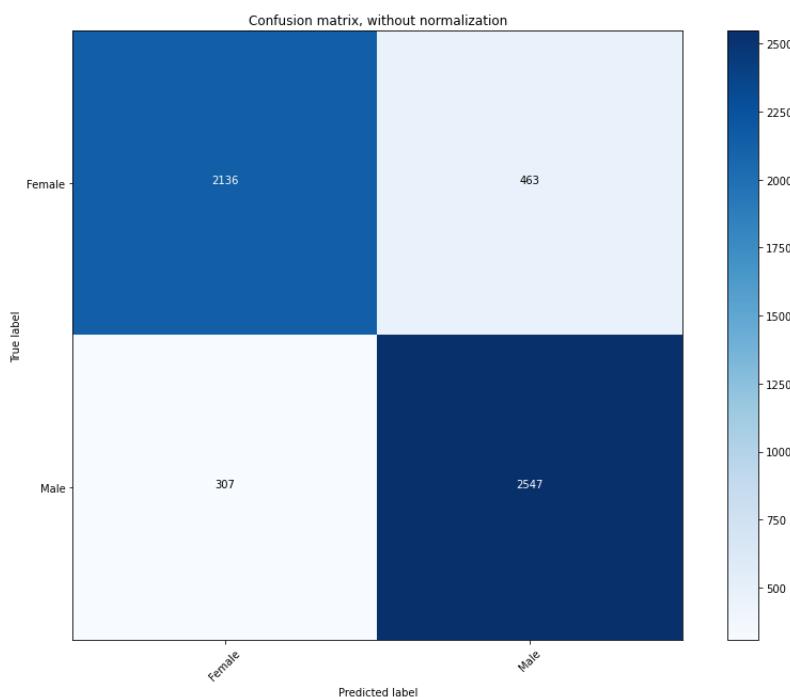
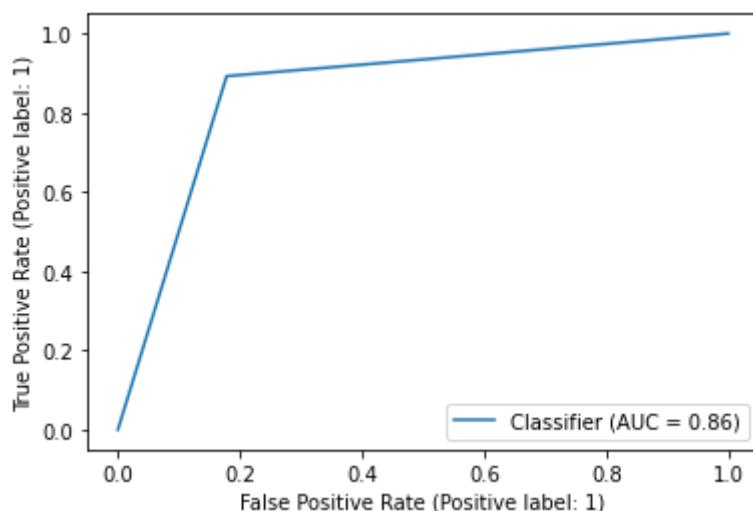
فصل ۳. طبقه بندی

- با توجه به نتیجه گیری بالا یکی از کارهایی که در قسمت پیش پردازش داده‌ها می‌توان انجام داد تا عملکرد طبقه بندی بهبود یابد، فیلتر کردن داده‌ها توسط یک فیلتر میان گذر در حوزه فرکانس خواهد بود
- با توجه به موارد بالا یکی از روش‌هایی که می‌توان برای طبقه بندی استفاده کرد استفاده از روش یکی در مقابل همه است. از آن جایی که کلاس عصبانیت در مجموع دقت بهتری نسبت به بقیه دارد و کلاس خنثی دقت پایین‌تری نسبت به بقیه دارد پس در ابتدا کلاس عصبانیت را طبقه بندی کرده، سپس کلاس خوشحالی در مرحله بعد کلاس ناراحتی و در آخر کلاس خنثی را طبقه بندی می‌کنیم.

در این قسمت نتایج طبقه بندی بر روی جنسیت را نمایش می دهیم و در پایان تحلیل مختصری از نتایج بدست آمده ارائه می کنیم. دقت شود توضیحات مربوط به طبقه بندها و پارامترهای آنها در قبل آمده است و از گفتن دوباره پرهیز می کنیم:

۳.۴.۳ اعمال KNN برای طبقه بندی "جنسیت" با استفاده از دسته ویژگی MFCC

با استفاده از این روش به دقت کلی ۰.۸۶ رسیده ایم. دیگر نتایج بدست آمده به صورت زیر است:

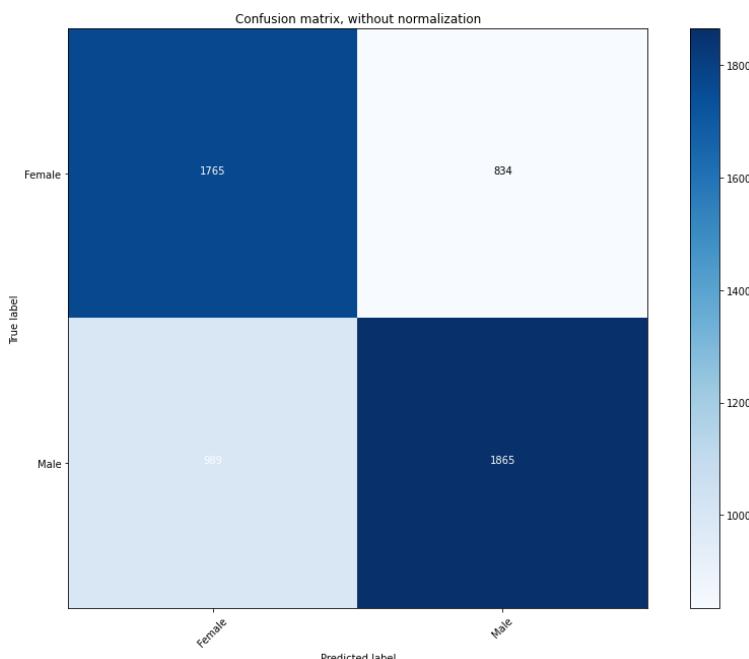
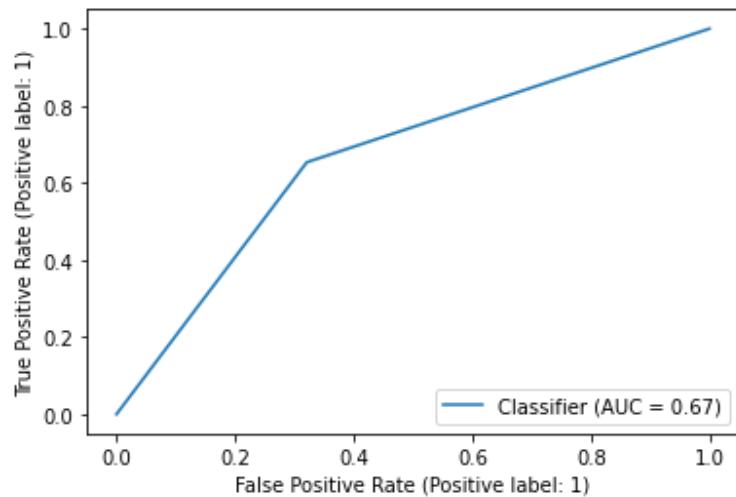


classes	Recall	Precision	F1-Score
0 female	0.82	0.87	0.84
1 male	0.89	0.84	0.86

- کلاس مرد با دقت بالاتری نسبت به کلاس زن طبقه بندی شده است.
- ۵۵ درصد از کل نمونه ها به کلاس مرد نسبت داده شده اند یعنی کلاسیفایر KNN در اینجا کلاس مرد را با احتمال بیشتری نسبت به کلاس زن طبقه بندی می کند.

۴.۴.۳ اعمال KNN برای طبقه بندی "جنسیت" با استفاده از دسته ویژگی :Chroma Frequency

با استفاده از این روش به دقت کلی ۰.۶۷ رسیده ایم. دیگر نتایج بدست آمده به صورت زیر است:

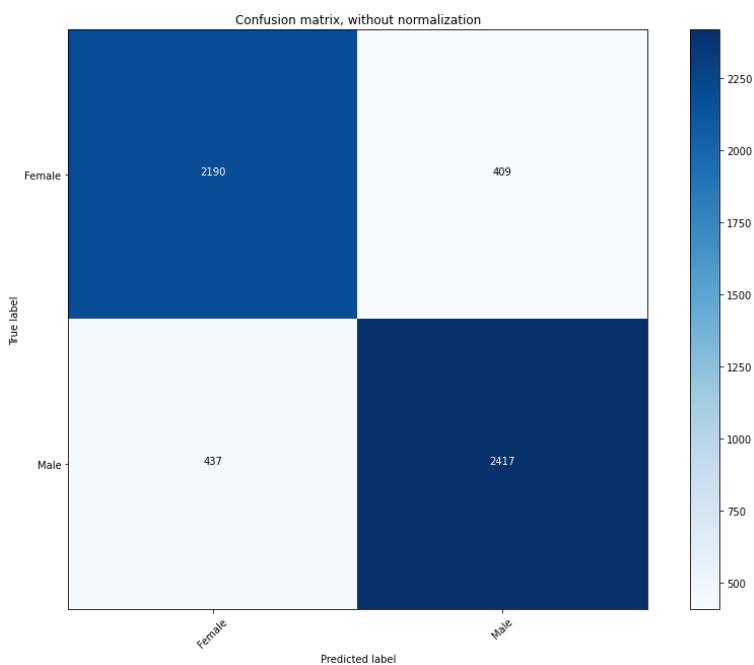
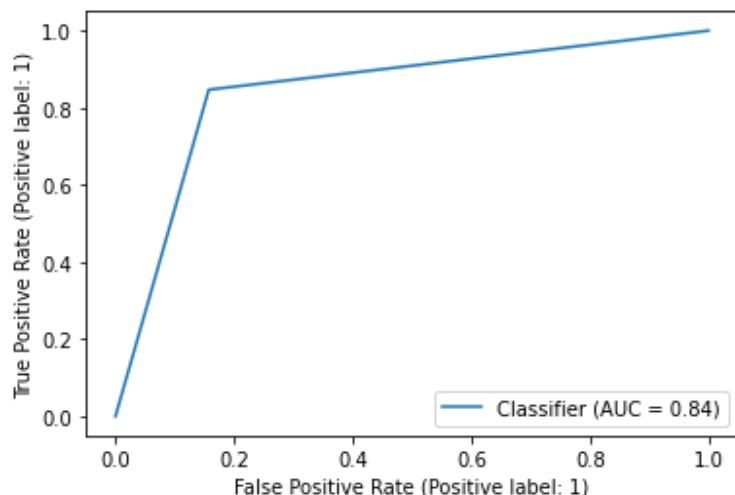


classes		Recall	Precision	F1-Score
0	female	0.68	0.64	0.66
1	male	0.65	0.69	0.67

- بر خلاف حالت قبلی، در اینجا کلاس زن با دقت بالاتری نسبت به کلاس مرد طبقه بندی شده است.
- با مقایسه نتایج KNN روی دسته ویژگی‌های Chroma و MFCC می‌توان گفت نتایج طبقه بندی بر روی MFCC به وضوح بهتر از Chroma است. درواقع قابلیت تفکیک پذیری دسته ویژگی‌های MFCC بیشتر بوده است و در نتیجه دقت بیشتری داشته است.

۵.۴.۳ اعمال MLP برای طبقه بندی "جنسیت" با استفاده از دسته ویژگی MFCC

با استفاده از این روش به دقت کلی ۰.۸۴ رسیده ایم. دیگر نتایج بدست آمده به صورت زیر است:

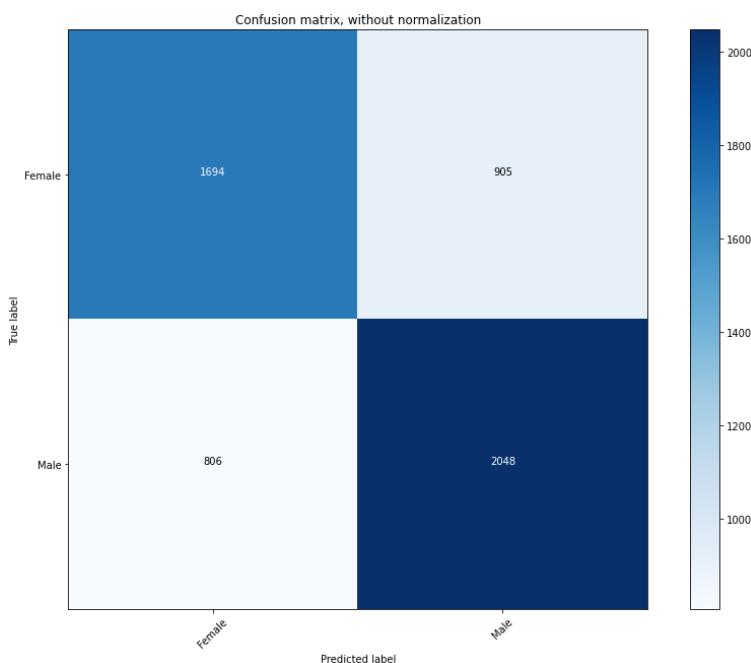
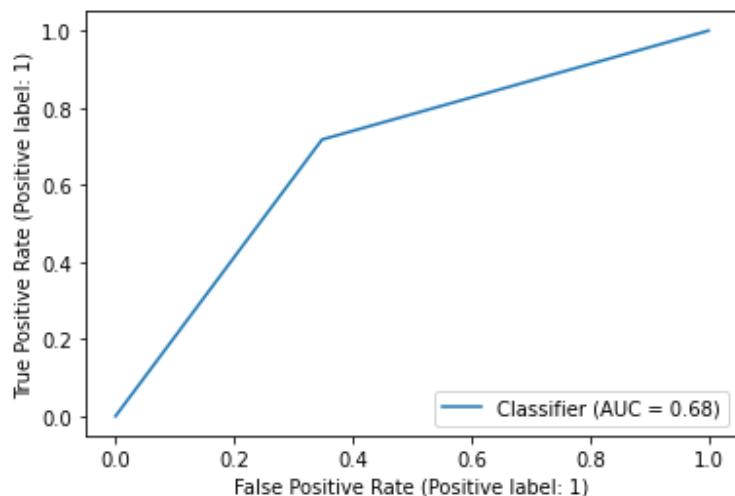


	classes	Recall	Precision	F1-Score
0	female	0.84	0.83	0.84
1	male	0.85	0.86	0.85

- در اینجا همانند طبقه بند KNN کلاس مرد با دقت بالاتری نسبت به کلاس زن طبقه بندی شده است.
- دقت طبقه بند MLP کمتر از طبقه بند KNN بدست آمده است.

۶.۴.۳ اعمال MLP برای طبقه بندی "جنسیت" با استفاده از دسته ویژگی :Chroma Frequency

با استفاده از این روش به دقت کلی ۰.۶۹ رسیده ایم. دیگر نتایج بدست آمده به صورت زیر است:



classes		Recall	Precision	F1-Score
0	female	0.65	0.68	0.66
1	male	0.72	0.69	0.71

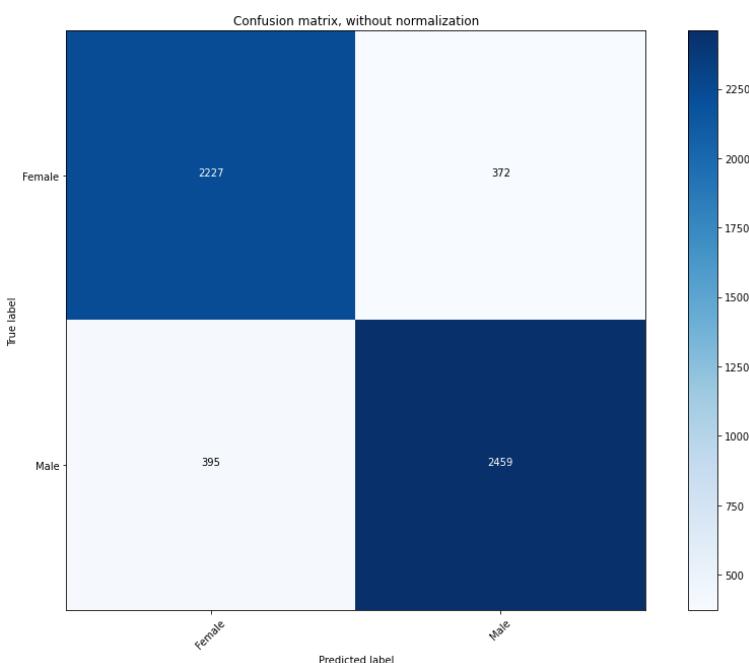
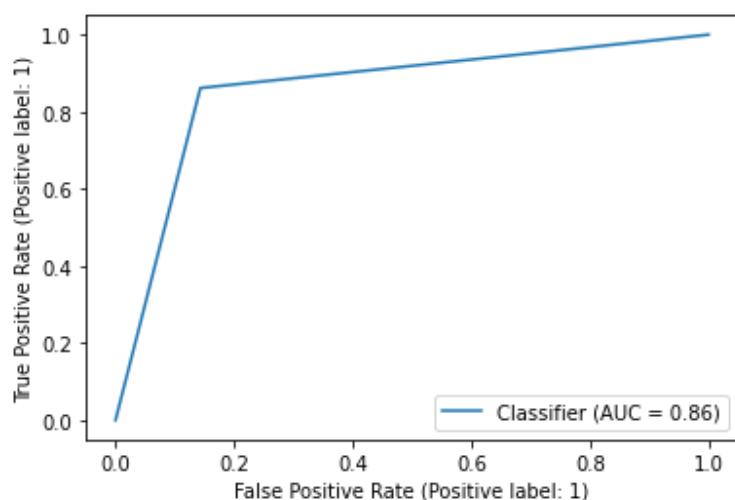
- کلاس مرد با دقت بالاتری نسبت به کلاس زن طبقه بندی شده است.

• ۵۴ درصد از کل نمونه ها به کلاس مرد نسبت داده شده اند یعنی کلاسیفایر MLP در اینجا کلاس مرد را با احتمال بیشتری نسبت به کلاس زن طبقه بندی می کند.

- دقت MLP روی دسته ویژگی Chroma از دقت KNN بیشتر بوده است.

۷.۴.۳ اعمال SVM برای طبقه بندی "جنسیت" با استفاده از دسته ویژگی MFCC

با استفاده از این روش به دقت کلی ۰.۸۶ رسیده ایم. دیگر نتایج بدست آمده به صورت زیر است:



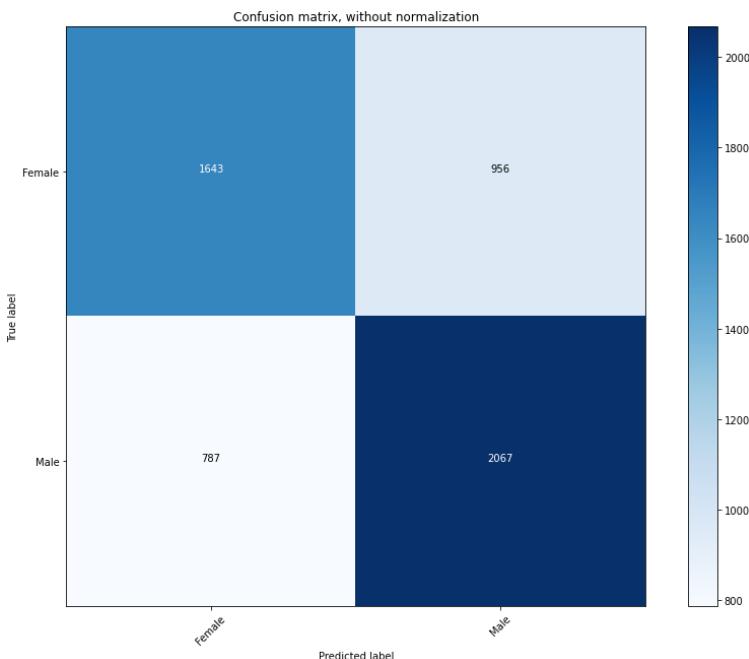
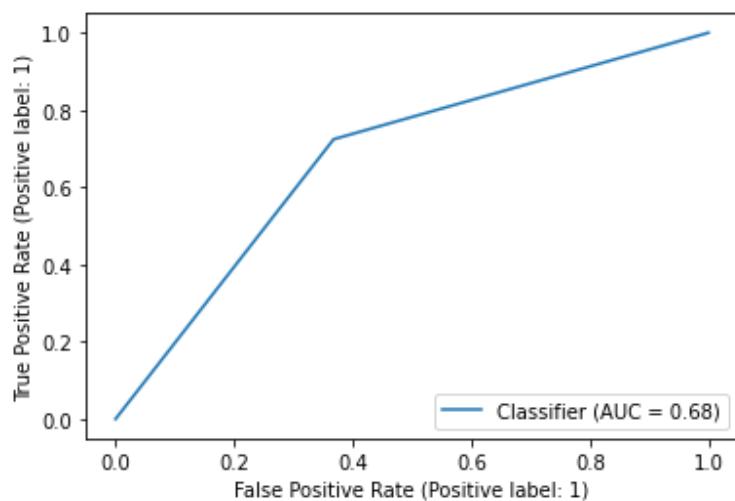
	classes	Recall	Precision	F1-Score
0	female	0.86	0.85	0.85
1	male	0.86	0.85	0.87

- کلاس مرد و زن با دقت یکسان طبقه بندی شده اند.

- دقت SVM روی دسته ویژگی MFCC از دقت MLP بیشتر بوده و با KNN یکسان است.

۸.۴.۳ اعمال MLP برای طبقه بندی "جنسیت" با استفاده از دسته ویژگی Chroma Frequency

با استفاده از این روش به دقت کلی ۰.۶۸ رسیده ایم. دیگر نتایج بدست آمده به صورت زیر است:



	classes	Recall	Precision	F1-Score
0	female	0.63	0.68	0.65
1	male	0.72	0.68	0.7

- کلاس مرد با دقت بیشتری نسبت به زن طبقه بندی شده است.
- دقت SVM روی دسته ویژگی Chroma از دقت MFCC روی دسته ویژگی های KNN بیشتر بوده و با KNN یکسان است.
- ۵۶ درصد از کل نمونه ها به کلاس مرد نسبت داده شده اند یعنی کلاسیفایر MLP در اینجا کلاس مرد را با احتمال بیشتری نسبت به کلاس زن طبقه بندی می کند

جمع بندی نهایی نتایج طبقه بندی جنسیت با استفاده از دسته ویژگی های Chroma و MFCC

خلاصه نتایج بدست آمده توسط کلاس بندی Chroma Frequency و MFCC بر روی دو دسته ویژگی SVM و MLP و KNN به صورت زیر است :

	MFCC				
	classes	KNN	MLP	SVM	Average
0	Accuracy of female	0.82	0.85	0.86	0.84
1	Accuracy of male	0.89	0.84	0.86	0.87
2	totala Accuracy	0.86	0.85	0.86	0.85

	Chroma Frequency				
	classes	KNN	MLP	SVM	Average
0	Accuracy of female	0.68	0.65	0.63	0.65
1	Accuracy of male	0.65	0.72	0.72	0.70
2	totala Accuracy	0.67	0.69	0.68	0.68

- همان طور که مشاهده می شود دقت کلی هر یک از طبقه بندها روی دسته ویژگی های Chroma بیشتر از دسته ویژگی های MFCC بوده است به طوری که به طور متوسط دقت روی MFCC حدود ۱۸ درصد بالاتر از دقت روی Chroma بوده است.
- در مجموع کلاس مرد در هر دو دسته ویژگی با دقت بالاتری نسبت به کلاس زن دسته بندی شده است
- از آن جایی که نتایج طبقه بندی بر روی Chroma به وضوح بیشتر از MFCC است درواقع قابلیت تفکیک پذیری MFCC نسبت به Chroma بیشتر بوده است و در نتیجه دقت بیشتری داشته است از آنجایی که MFCC در حوزه فرکانس، ویژگی استخراج می کند می توان گفت که داده ها در حوزه فرکانس تفکیک پذیری بیشتری دارند

پیوست: به صورت کلی نتایج طبقه بندی بدست آمده برای تشخیص احساسات از روی داده های موجود از دقت بالایی برخودار نیستند و از دلایل آن میتوان به نامناسب بودن نمونه صدای های ضبط شده اشاره داشت که تشخیص آنها برای انسان نیز کاری مشکل است و نمیتوان با دقت بالایی تشخیص داد. از این جهت علاوه بر طبقه بندی های انجام شده و ذکر شده در بالا تلاش کردیم تا با روش های مختلف دیگری نیز به طبقه بندی داده ها بپردازیم تا شاید نتایج بیشتری دریافت شود. برای این امر از روش های زیر بهره گرفتیم:

Forward feature selection •

Bagging •

Adaboost •

Naiive Bayes •

Regression •

اما در نتایج دریافتی از هر یک از این طبقه بندها رشد یا پیشرفتی در دقت مشاهده نشد. برای مثال نتایج Ensemble Learning که به کمک Accuracy: 0.39 انجام شده به شرح زیر است:

classes		Recall	Precision	F1-Score
0	Angry	0.41	0.41	0.41
1	Sad	0.37	0.39	0.38
2	Happy	0.44	0.38	0.41
3	Neutral	0.32	0.36	0.34

همانطور که مشاهده می شود علی رغم انتظاری که از ماهیت روش های ensemble داشتیم افزایش دقیقی در این روش نسبت به روش بهینه ما که svm است مشاهده نمی شود. باقی طبقه بندی های بررسی شده نیز مشابه همین مورد، پیشرفتی را حاصل ننموده اند و میتوان نتیجه گرفت که با کمک فیچرهای انتخاب شده و در دسترس ما و این نکته که برای طبقه بندی ها از gridsearch استفاده شده و مقادیر پارامترها به صورت بهینه بدست آمده اند، بهترین نتیجه همان نتایج بدست آمده برای svm و ویژگی های MFCC می باشد.

فصل ۴

خوشه‌بندی

فرایند خوشه‌بندی یا همان clustering را به زبان ساده می‌توان اینگونه بیان کرد که هدف اختصاص object‌ها به دسته‌ها و یا خوشه‌ها است بطوریکه شباهت اشیاء داخل یک دسته حداکثر و شباهت اشیاء در دسته‌های مختلف حداقل باشد. کلاسترینگ در مباحث ماشین لرنینگ جز روش‌های unsupervised تلقی می‌شود، چراکه در این روش کاری به برچسب و لیبل داده‌ها نداریم و هدف یافتن نظم داده‌ها و توصیف ساختار پنهان آنهاست. متدهای فراوانی برای انجام clustering وجود دارد که در این پژوهه قصد داریم به اعمال و تحلیل دو الگوریتم

clustering بر روی داده‌های موجود پردازیم، یکی با استفاده از الگوریتم K-Means و دیگری با استفاده از روش Agglomerative clustering یکی از موارد مهم و هائز اهمیتی که قبل از خوشه‌بندی داده‌ها باید بدایمن این است که عموماً الگوریتم‌های clustering با داده‌هایی با ابعاد بالا به خوبی کار نمیکنند، چرا که به خاطر مسئله curse of dimensionality زمانی که ابعاد فضای زیاد می‌شود، فاصله بین نقاط داده در چنین فضایی کم می‌شود و این یک مشکل بزرگ برای clustering بحساب می‌آید به ویژه زمانی که با الگوریتم‌هایی سر و کار داریم که مانند distance-base اصطلاحاً هستند و از معیارهای مبتنی بر فاصله برای شناسایی نقاط مشابه و خوشه‌بندی آن‌ها استفاده می‌کنند. در نتیجه، لازم است که ابعاد فضایا به بیان بهتر ابعاد داده‌هایمان را کم کنیم. از طرفی چون در این پژوهه سعی داریم تا پس از انجام مرحل خوشه‌بندی داده‌های clustering visualize کنیم و نتیجه را تحلیل کنیم، پس بهتر است ابعاد داده‌ها یمان را که اکنون ۱۲ تا می‌باشد به ۲ بعد کاهش دهیم تا بتوانیم به بهترین و قابل درک ترین شکل خوشه‌بندی را تحلیل کنیم. اما یک نکته خیلی مهم وجود دارد، اینکه کاهش بعد می‌تواند خوشه‌بندی را کاهش بدهد، به بیان دیگه ممکن باعث شود داده‌ها با هم همپوشانی داشته باشند که در این صورت نمیتوان آن‌ها را به درستی خوشه‌بندی کرد. در نتیجه لازم است تا بتدا از قبل از خوشه‌بندی، بهترین روشی که میتوان برای dimension reduction استفاده کرد را ببابیم.

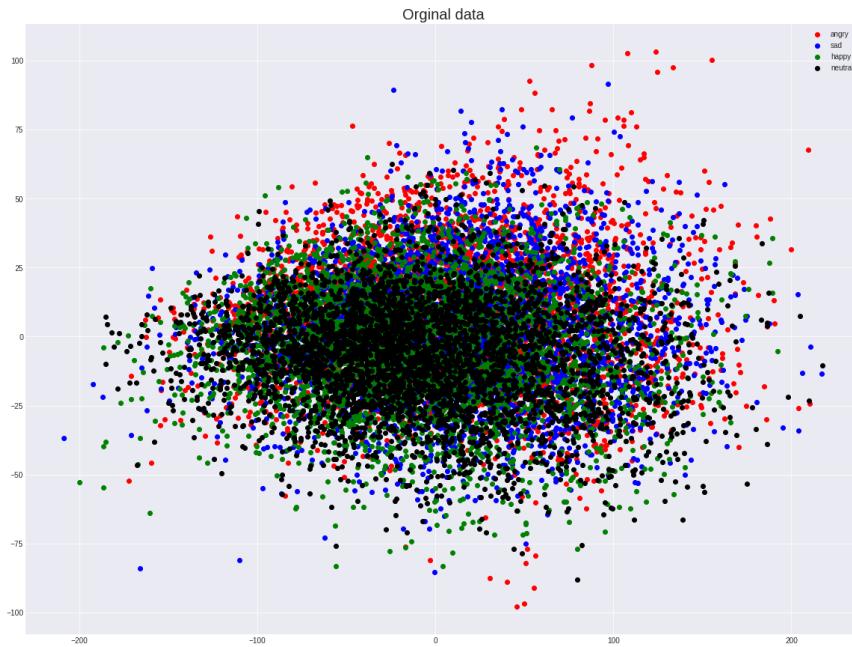
در اینجا ما از ۴ روش کاهش بعد استفاده کرده و ابعاد داده را به ۲ بعد کاهش می‌دهیم. سپس داده‌ای که پس از کاهش بعد تفکیک پذیری بهتری داشته باشد را انتخاب و کلاسترینگ روی آن اعمال می‌کنیم.

حال معرفی مختصری از روش‌های کاهش بعد انجام شده خواهیم داد و داده‌ها را توسط هریک کاهش بعد می‌دهیم:

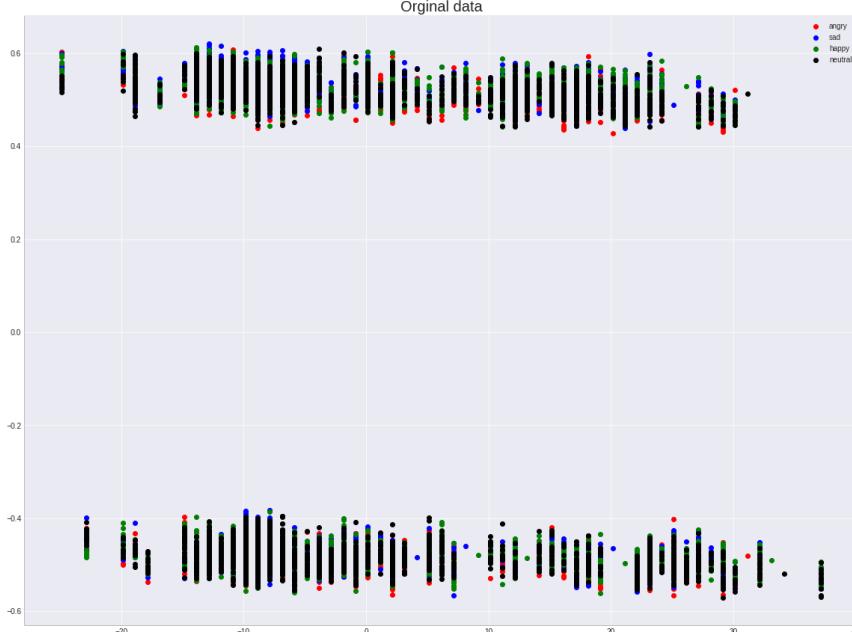
PCA ۱.۴

Principal Component Analysis یا به اختصار PCA یکی از ساده‌ترین و تاحد زیادی رایج‌ترین روش‌ها برای کاهش ابعاد است و می‌توان آن را به عنوان یک روش فشرده سازی در نظر گرفت که به صورت Linear ابعاد را ترکیب می‌کند تا آنها را کاهش دهد. ویژگی اصلی PCA آن است که تا آنجا که ممکن است variance مجموعه داده‌ها را حفظ می‌کند.

با اعمال PCA روی دسته ویژگی‌های MFCC نتیجه زیر حاصل شده است:



با اعمال PCA روی دسته ویژگی‌های Chroma نتیجه زیر حاصل شده است:



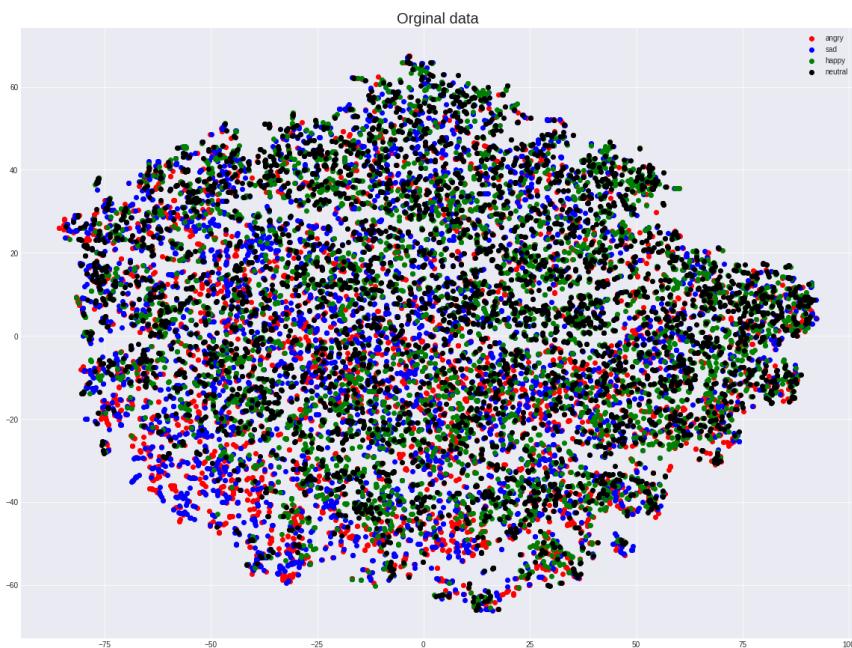
مشاهده می شود که دسته ویژگی‌های Chroma پس از کاهش به همپوشانی بسیار زیاد و تفکیک پذیری بسیار کم دارند برای دسته ویژگی MFCC شرایط کمی بهتر است ولی بازهم داده‌ها همپوشانی زیادی دارند.

t-SNE ۲.۴

یا همان t-SNE یک الگوریتم یادگیری ماشین برای تجسم داده‌های با ابعاد بالا است که توسط Geoffrey Hinton و Laurens van der Maaten در ۲۰۱۸ جایزه تورینگ و دریافت

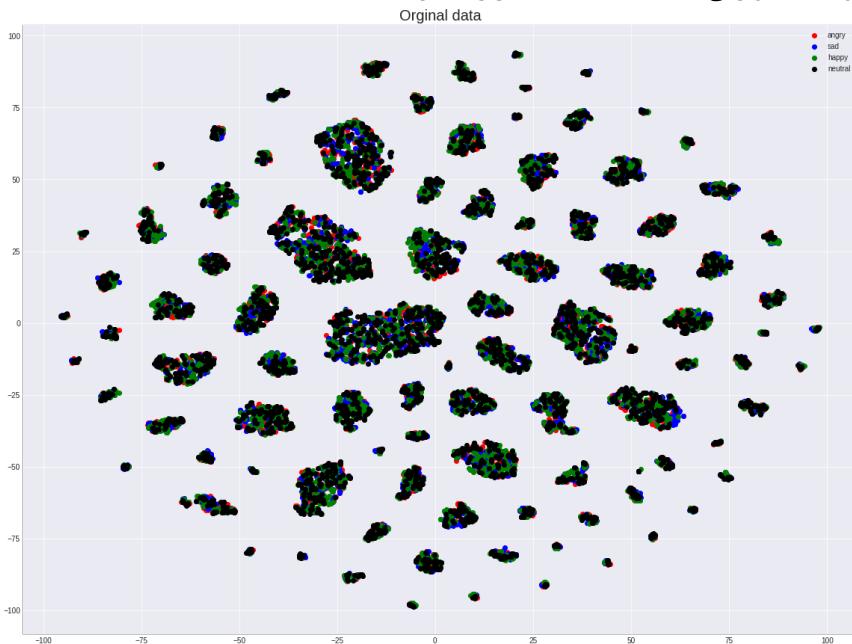
کرد) پیشنهاد شده است. این تصور وجود دارد که داده‌های طبیعی با ابعاد بالا، در یک منیفولد با ابعاد پایین در فضایی با ابعاد بالا قرار دارند. هدف t-SNE حفظ توبیولوژی این منیفولد با حفظ فواصل محلی هنگام حرکت از ابعاد بالا به پایین است.

با اعمال t-SNE روی دسته ویژگی‌های MFCC نتیجه زیر حاصل شده است:



مشاهده می‌شود باز هم همپوشانی زیاد و حتی بدتر از حالت قبلی یعنی استفاده از PCA است.

با اعمال t-SNE روی دسته ویژگی‌های Chroma نتیجه زیر حاصل شده است:

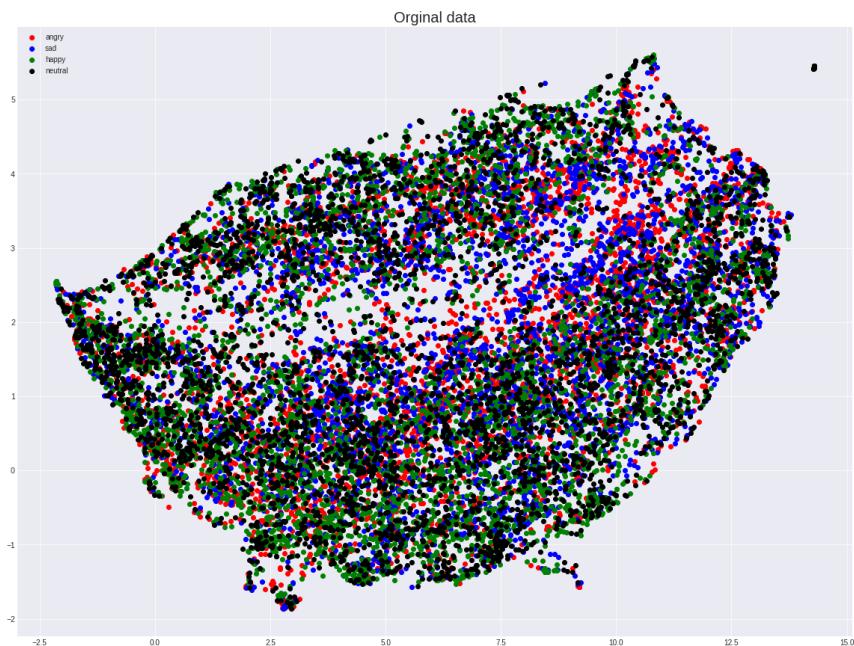


همانطور که در بخش قبلی مشاهده کردیم دسته ویژگی‌های Chroma تفکیک پذیری کمی دارند؛ درنتیجه پس از کاهش بعد، همپوشانی سیاست زیادی دارند.

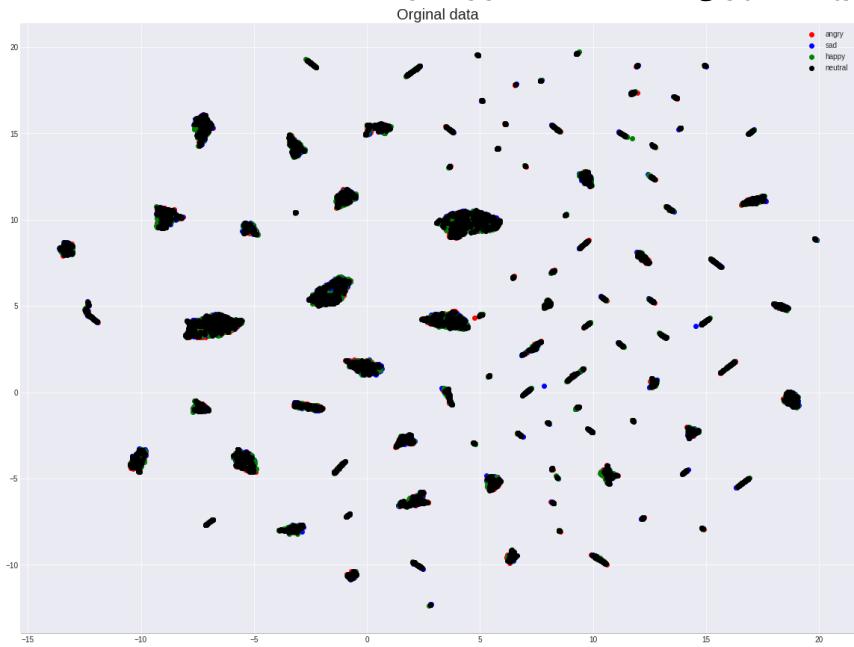
UMAP ۳.۴

UMAP به گفته نویسنده‌گانش از منظر کیفیت t-SNE visualization رقابت میکند اما یک تفاوت خیلی مهم دارد آن هم اینکه UMAP بیشتر ساختار کلی داده را حفظ می‌کند به طوری که فاصله بین خوشه‌های تشکیل شده در t-SNE معنی قابل توجهی ندارد، اما در UMAP فاصله بین خوشه‌ها مهم است.

با اعمال UMAP روی دسته ویژگی‌های MFCC نتیجه زیر حاصل شده است:



با اعمال UMAP روی دسته ویژگی‌های Chroma نتیجه زیر حاصل شده است:

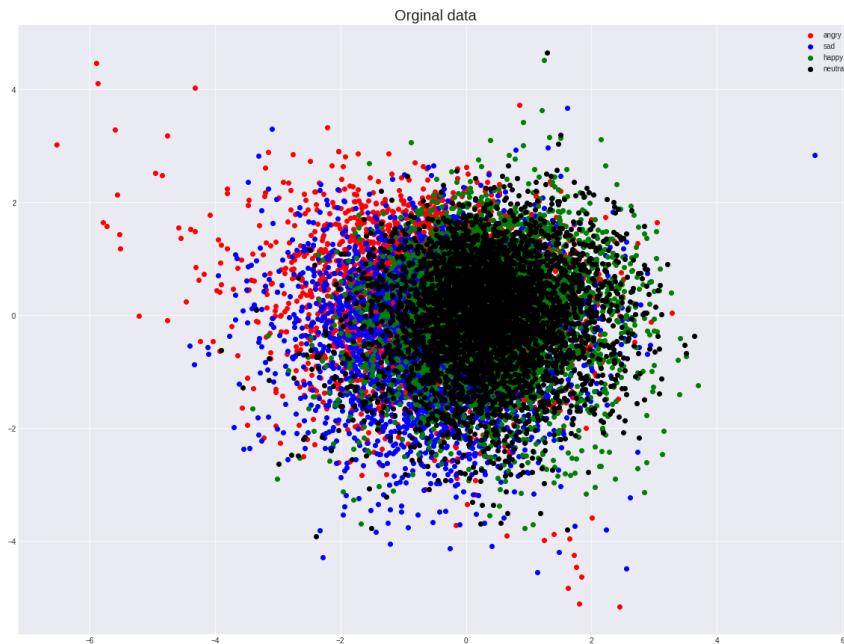


مشاهده می‌شود شرایط نسبت به دو حالت قبلی بدتر شده است و تفکیک پذیری داده‌ها کمتر شده است.

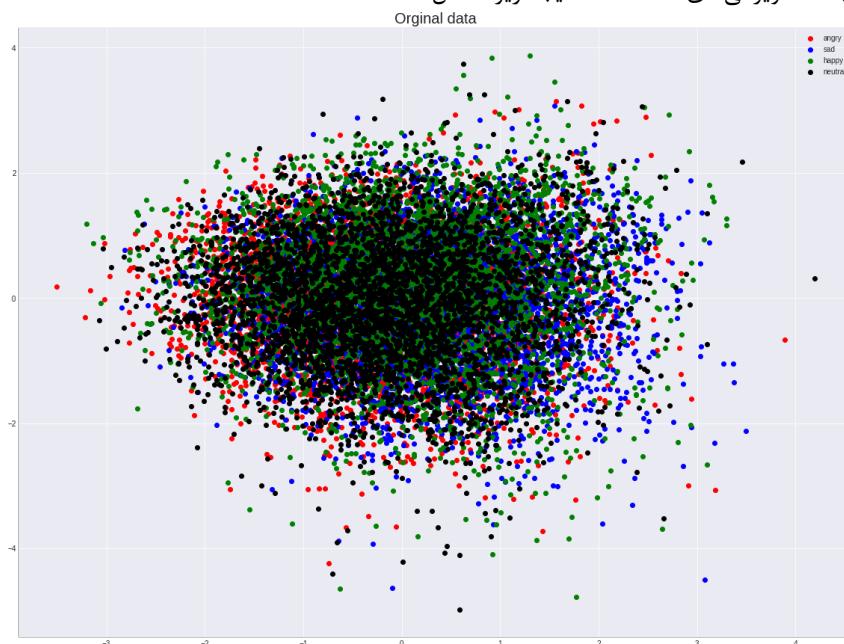
LDA ۴.۴

روی مجموعه داده است. به طوری که بتواند نسبت پراکندگی بین کلاس‌ها (Between Class) را به پراکندگی درون کلاس‌ها (Within Class) تبدیل شده، حداقل کند.

با اعمال LDA روی دسته ویژگی‌های MFCC نتیجه زیر حاصل شده است:



با اعمال LDA روی دسته ویژگی‌های Chroma نتیجه زیر حاصل شده است:



در مقایسه با سه الگوریتم قبلی، داده‌های کاهش بعد یافته توسط LDA کمی تفکیک پذیری بهتری دارند ولی همچنان همپوشانی داده‌ها

زیاد است و نباید انتظار کلاسترینگ خوبی داشته باشیم.

درنهایت در ادامه کار ما داده‌های کاهش بعدیافته توسط الگوریتم LDA را خوشبندی می‌کنیم.

۵.۴ معرفی روش‌های کلاسترینگ

ما در این پژوهه از دو الگوریتم استفاده می‌کنیم. اما قبل از اینکه بخواهیم تصمیم بگیریم که از چه الگوریت‌هایی می‌توان استفاده کرد، باید توجه کنیم که مطابق تعریف پژوهه لازم است تا با تعداد cluster های متفاوت داده‌ها را دسته بندی کنیم، این بدان معناست که نمی‌توانیم از روش‌های OPTICS و یا DBSCAN density-based ای مانند کنیم. چرا که در این الگوریتم‌ها نمی‌توان تعداد کلاستر های را مشخص کرد.

K-Means ۱.۵.۴

یک الگوریتم k-means distance-based می‌باشد که به صورت iterative سعی در یافتن بهترین نقاط به عنوان مرکز خوش‌ها را دارد. این الگوریتم همواره با نقص‌هایی همراه است برای مثال فقط هنگامی که خوش‌ها کروی فرض شوند بهترین عملکرد خوش رو خواهد داشت یا ممکن است به خاطر ماهیت local minimum iterative خود به منجر به نتیجه نادرست شود؛ اما نکته اینجاست که با همه این تفاسیر همچنان k-mean از الگوریتم‌های محبوب و پرکاربرد به حساب می‌آید لذا ما در اینجا از آن استفاده می‌کنیم.

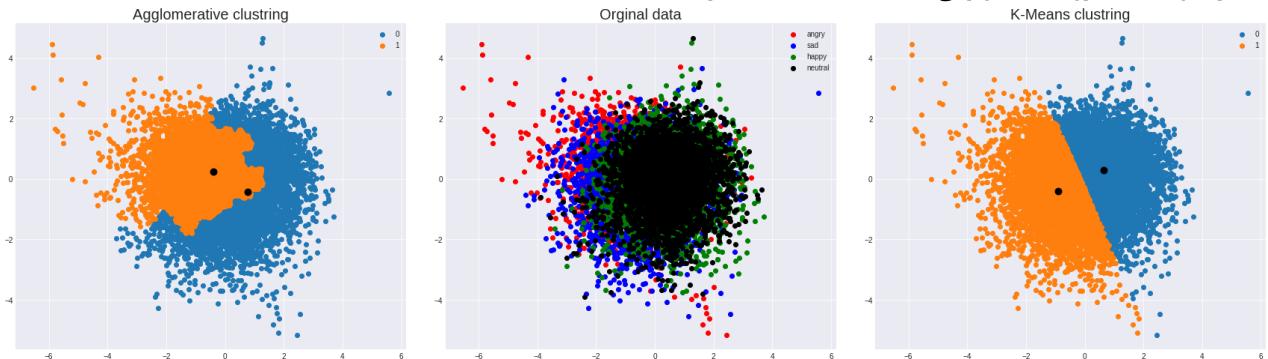
Agglomerative Hierarchical Clustering ۲.۵.۴

این الگوریتم از رایج‌ترین روش hierarchical clustering می‌باشد و برای گروه بندی اشیا در خوش‌ها بر اساس شباهت استفاده می‌شود، هرچه اشیاء به هم نزدیک تر باشند، بهم شبیه تر خواهند بود، البته نزدیک تر بودن بر اساس linkage انتخابی در این الگوریتم، می‌تواند تفاسیر مختلفی داشته باشد. ما برای این پژوهه از complete linkage استفاده می‌کنیم، یعنی نزدیک ترین اشیاء به هم آنها می‌هستند که فاصله آنها از هم، کمتر از maximum فاصله سایر (جفت) اشیاء موجود باشد. یعنی اینجا شباهت یک معیار نسبی تلقی می‌شود. در این الگوریتم هر شی در ابتدا به عنوان یک خوش‌هه تک عنصری در نظر گرفته می‌شود و در هر مرحله، دو خوش‌هه که بیشترین شباهت را دارند، در یک خوش‌هه بزرگتر جدید ترکیب می‌شوند. این روش تا زمانی تکرار می‌شود که همه نقاط عضو تنها یک خوش‌هه بشوند.

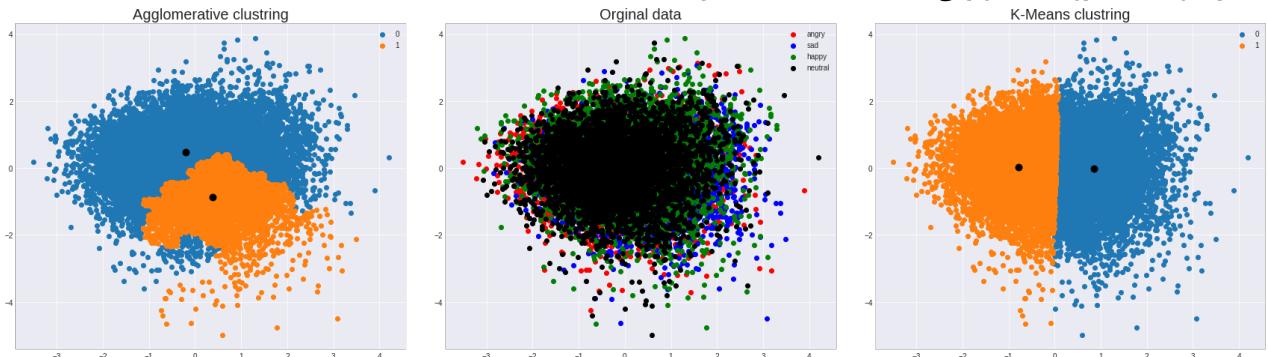
۶.۴ نتایج خوشبندی:

2- Cluster ۱.۶.۴

اعمال خوشبندی روی دسته ویژگی های MFCC که کاهش بعد یافته اند :



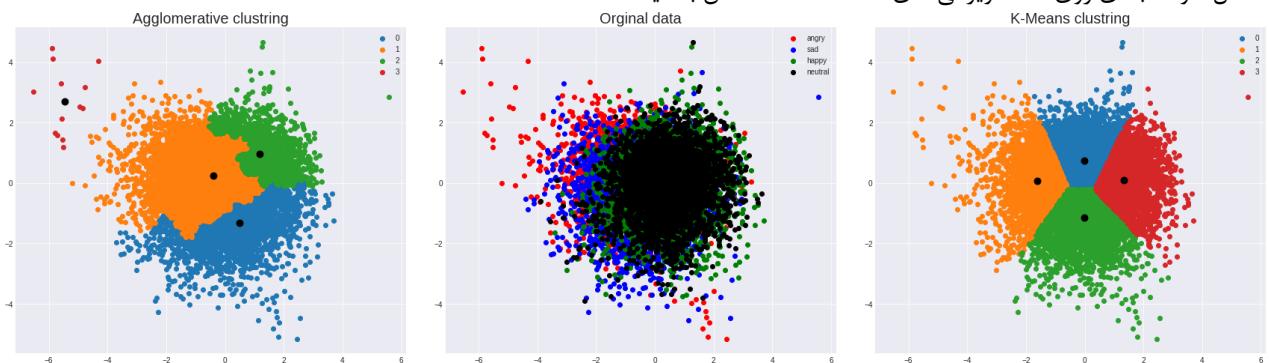
اعمال خوشبندی روی دسته ویژگی های Chroma که کاهش بعد یافته اند :



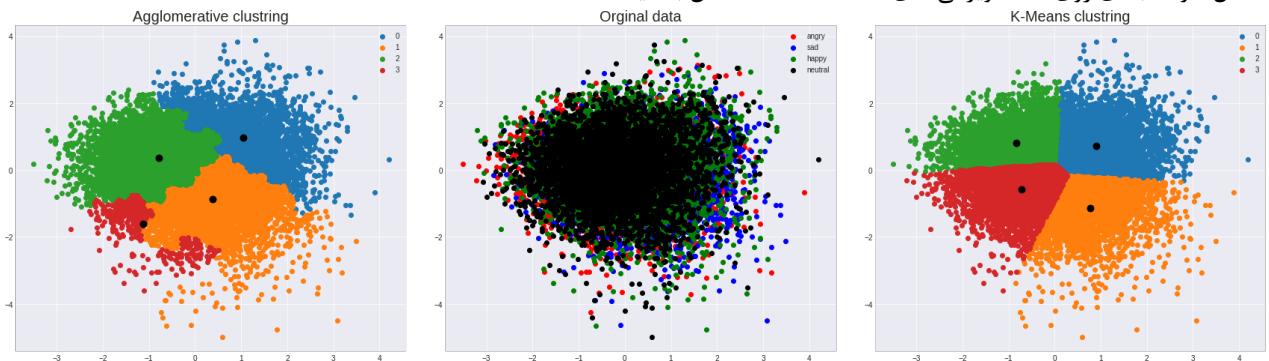
در ابتدا دقت شود از آن جایی که داده ها همپوشانی بسیار زیای دارند به طوری که حتی با استفاده از روش های یادگیری با ناظر که در فصل قبل انجام شد نمی توان داده ها را با دقت مناسبی تفکیک کرد بلطفاً نمی توان انتظار داشت که بتوان با استفاده از روش های خوشبندی داده کلاس ها را تفکیک کرد. با مقایسه دو نمودار بالا می توان گفت که کلاس عصبانیت توسط خوشبندی به روش Agglomerative روی دسته ویژگی های MFCC تا حدی جدا شده است نتیجه ای که در فصل قبل هم گرفته بودیم که کلاس عصبانیت با دقت بیشتری نسبت به بقیه دسته بندی شده بود. اما استفاده از k-means در این حالت نتوانسته عملکرد خوبی داشته باشد. دقت شود داده های مربوط به ویژگی های Chroma به شدت همپوشانی زیادی دارند و انتظار نمی رود که بتوان آن ها را به خوبی جدا کرد.

4- Cluster ۲.۶.۴

اعمال خوشه‌بندی روی دسته ویژگی‌های MFCC که کاهش بعد یافته اند :



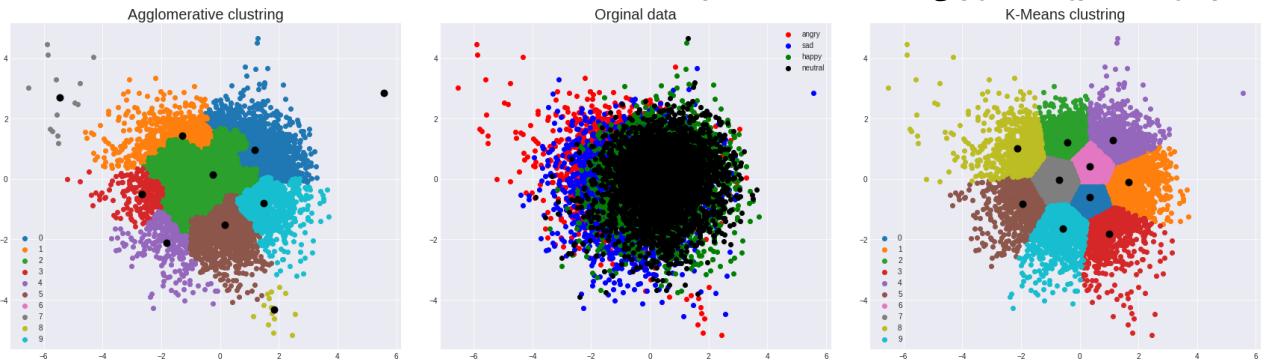
اعمال خوشه‌بندی روی دسته ویژگی‌های Chroma که کاهش بعد یافته اند :



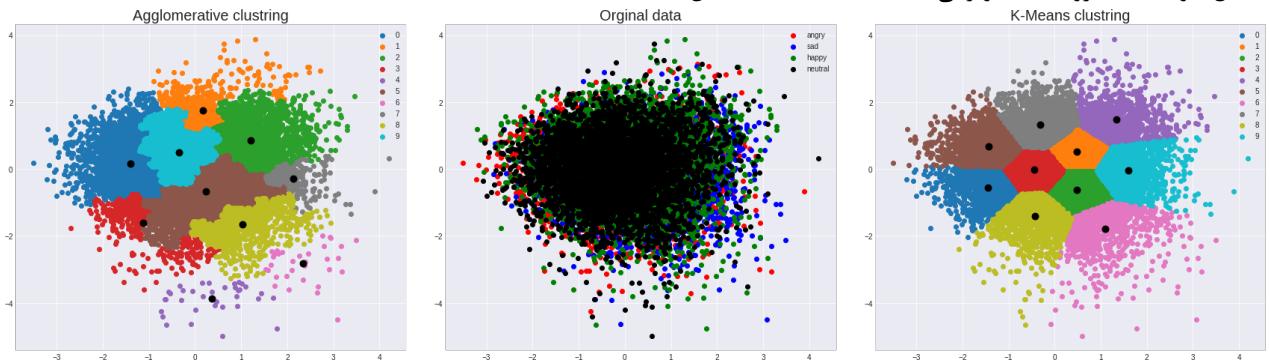
در ابتدا دقت شود از آن جایی که داده‌ها همپوشانی بسیار زیای دارند به طوری که حتی با استفاده از روش‌های یادگیری با ناظر که در فصل قبل انجام شد نمی‌توان داده‌ها را با دقت مناسبی تفکیک کرد فلذًا نمی‌توان انتظار داشت که بتوان با استفاده از روش‌های خوشه‌بندی داده کلاس‌ها را تفکیک کرد. با مقایسه دو نمودار بالا می‌توان گفت که کلاس عصبانیت توسط خوشه‌بندی به روش Agglomerative روی دسته ویژگی‌های MFCC تا حدی جدا شده است نتیجه‌ای که در فصل قبل هم گرفته بودیم که کلاس عصبانیت با دقت بیشتری نسبت به بقیه دسته بندی شده بود. اما استفاده از k-means در این حالت نتوانسته عملکرد خوبی داشته باشد. دقت شود داده‌های مربوط به ویژگی‌های Chroma به شدت همپوشانی زیادی دارند و انتظار نمی‌رود که بتوان آن‌ها را به خوبی جدا کرد.

10- Cluster ۳.۶.۴

اعمال خوشبندی روی دسته ویژگی های MFCC که کاهش بعد یافته اند :



اعمال خوشبندی روی دسته ویژگی های Chroma که کاهش بعد یافته اند :



در این حالت با توجه به پراکندگی داده های اصلی به نظر می رسد که کلاس عصبانیت مربوط به دسته ویژگی های MFCC در سه کلاستر(زرد و خاکستری و نارنجی) توسط روش Agglomerative تاحدی جدا شده است و داده های مربوط به کلاس خوشحالی نیز در دو کلاستر(آبی) تا حد کمی جدا شده است همچنین کلاس عصبانیت مربوط به دسته ویژگی های MFCC در یک کلاستر(زرد) توسط روش k-means تا حدی جدا شده است. برای دسته ویژگی های Chroma نیز می توان گفت که کلاس ناراحتی دو کلاستر(زرد و خاکستری) توسط روش Agglomerative تا حدی جدا شده اند و توسط روش k-means در سه کلاستر جدا شده اند که تا حدی با توجه به نتایج بدست آمده در فصل قبلی قابل توجیه است. همچنین کلاس عصبانیت نیز به نظر می رسد که در دو کلاستر(آبی و قرمز) توسط روش Agglomerative و همچنین در دو کلاستر(قهوه ای و آبی) توسط روش Agglomerative تا حد کمی جدا شده اند.

۷.۴ جمع بندی نهایی :

کلاس خنثی بیشترین همپوشانی شدیدی را با بقیه کلاس‌ها داشته و در نتیجه به هیچ وجه نمی‌توان به درستی آن را خوشه‌بندی کرد. کلاس عصبانیت در دسته ویژگی‌های MFCC تا حدی توسط خوشه‌بندی جدا شده است و همچنین کلاس نازاحتی و عصبانیت توسط دسته ویژگی‌های Chroma نیز تا حد کمی توسط خوشه‌بندی جدا شده‌اند که با نتایج بدست آمده در فصل قبل مطابقت دارد چرا که در فصل قبل نیز توسط روش‌های یادگیری با ناظر مشاهده کردیم که کلاس خنثی تفکیک‌پذیری کمی داشته و دقت طبقه‌بندی آن پایین تر از بقیه بود. با توجه به پراکندگی داده‌ها و درهم رفتگی و همپوشانی بسیار شدید داده‌های هر ۴ کلاس نمی‌توان انتظار خوشه‌بندی مناسب داشت. با توجه به اینکه دسته ویژگی‌های MFCC تفکیک‌پذیری بهتری نسبت به دسته ویژگی‌های Chroma دارند و از آنجایی که MFCC از ویژگی‌های فرکانسی استخراج می‌شود فلذًا می‌توان گفت که داده‌ها در حوزه فرکانس تفکیک‌پذیری بیشتری دارند بنابرین یکی از کارهایی که برای افزایش عملکرد مدل می‌توان انجام داد، فیلتر کردن داده‌ها توسط فیلتر میان‌گذر در حوزه فرکانس به عنوان یکی از بخش‌های پیش‌پردازش داده‌ها خواهد بود. با توجه به کمبود وقت و همچنین وقتگیر بودن پیدا کردن باند مناسب فیلترینگ ما از انجام این کار صرف نظر کردیم.