به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

درس یادگیری ماشین

گزارش پروژه پایانی

نام و نام خانوادگی	امیرحسین پورداود – امیرغرقابی – محمد ابوذری – مهدی سلیمانی زادگان
شماره دانشجویی	۸۱۰۱۰۱۱۲۰ – ۸۱۰۱۰۲۲۱۷ – ۸۱۰۱۰۱۰۸۷ – ۸۱۰۱۰۲۱۷۶
تاریخ ارسال گزارش	۱۴۰۲.۱۱.۰۴

فهرست

شکل‌ها

جدول‌ها

بخش ۱. پیش پردازش داده ها

۱.۱. تمیزکردن داده ها

پیش پردازش داده‌ها در داده‌های صوتی تأثیر قابل توجهی دارد. این فرآیند شامل مجموعه‌ای از تکنیک‌ها و الگوریتم‌ها است که به منظور بهبود کیفیت و قابلیت استفاده از داده‌های صوتی انجام می‌شود. تاثیرات اصلی پیش پردازش داده‌های صوتی عبارتند از:

Resample: ۱.۱.۱

ibrosa.resample یک تابع در کتابخانه‌ی librosa است که برای بازنمونده کردن یک سیگنال صوتی به یک نرخ نمونه‌برداری مختلف استفاده می‌شود. این تابع با گرفتن یک سیگنال صوتی و نرخ نمونه‌برداری مطلوب به عنوان ورودی، سیگنال صوتی بازنمونه‌برداری شده را برمی‌گرداند. استفاده از این تابع در وظایف پردازش صوتی متداول است که نرخ نمونه‌برداری برای تحلیل یا مقایسه‌ی بعدی نیاز به تغییر دارد.

Trim .۲.۱.۱

تابع trim در پردازش داده‌های صوتی به معنی برش (کوتاه کردن) قسمت های ابتدایی و/یا انتهایی یک سیگنال صوتی است. این عملیات معمولاً برای حذف بخش‌های بی‌فایده یا ناکارآمد از سیگنال صوتی استفاده می‌شود و می‌تواند بهبودی در کارایی الگوریتم‌های پردازش صوتی و موارد دیگر ایجاد کند. عموماً می‌توان از تفاوت زمان شروع و پایانی یک سیگنال صوتی با توجه به معیارهای مشخصی (مانند آستانه بالا/پایین، توان، انرژی، آمارهای زمانی و فرکانسی و غیره) استفاده کرده و قسمت‌های غیرمربوط را حذف نمود. این عملیات برای حذف سکوت‌ها، نویزها و قسمت‌هایی از سیگنال صوتی که در تحلیل یا استفاده‌ی بعدی مورد نیاز نیستند، مفید می‌باشد.

Normalization .۳.۱.۱

نرمال‌سازی یک تکنیک متداول در پیش پردازش داده‌های صوتی است که به هدف تغییر مقیاس و مقادیر سیگنال صوتی ورودی به یک محدوده استاندارد استفاده می‌شود، معمولاً بین ۰ تا ۱ یا -۱ تا ۱. این فرآیند به تطبیق مقیاس و دامنه مقادیر ویژگی‌ها یا مقادیر ورودی کمک می‌کند، که برای الگوریتم‌های یادگیری ماشین و وظایف پردازش سیگنال مفید است.

در این مسئله سعی شده است داده های صوتی با استفاده از فرمول X-min / (max-min) مقایس بندی شوند.

PreemPhasis .۴.۱.۱

پری‌امفاسیس (Preemphasis) یک تکنیک است که در پیش پردازش داده‌های صوتی به کار می‌رود. این تکنیک برای بهبود بخش‌های خاص فرکانسی صدا و افزایش قابلیت درک و کیفیت کلی سیگنال استفاده می‌شود. هدف اصلی از پری‌امفاسیس، تاکید بر بخش‌های فرکانسی بالا و کاهش بخش‌های فرکانسی پایین می‌باشد.

در داده‌های صوتی، انرژی سیگنال معمولاً در محدوده‌ی فرکانسی پایین قرار دارد. با اعمال پری‌امفاسیس، ما از طریق تقویت بزرگی مولفه‌های فرکانسی بالا نسبت به مؤلفه‌های فرکانسی پایین، کیفیت سیگنال را بهبود می‌بخشیم. برای این کار، با اعمال فیلتر پاس‌بالا به سیگنال، فرکانس‌های بالاتر تقویت و فرکانس‌های کمتر ضعیف می‌شوند.

با اعمال پری‌امفاسیس ، می‌توانیم نسبت سیگنال به نویز را افزایش دهیم، تعادل طیفی را افزایش دهیم و عملکرد الگوریتم‌های پردازش گفتار بعدی مانند تشخیص گفتار یا شناسایی گوینده را بهبود بخشیم. اجزای فرکانس بالا تقویت شده می تواند به حفظ جزئیات مهم و کاهش تأثیر نویز یا اعوجاج در طول تجزیه و تحلیل گفتار یا انتقال کمک کند.

۲.۱. استخراج ویژگی داده ها

استخراج ویژگی‌ها در داده‌های صوتی یکی از مهمترین مراحل در پردازش سیگنال صوتی و برای دستیابی به نتایج دقیق در وظایف پردازش صوتی است. در واقع، استخراج ویژگی‌ها به معنای تبدیل سیگنال صوتی از فرمت زمانی به فرمت فضایی است، که داده‌های صوتی را قابل تحلیل و استفاده‌ی بیشتر می‌کند.

استخراج ویژگی‌ها به منظور بهبود دقت تشخیص گفتار و تشخیص صدا و همچنین بهبود عملکرد مدل‌های یادگیری ماشین مورد استفاده قرار می‌گیرد. بسته به وظایف مورد نظر، ویژگی‌های مختلفی مانند: طول پالس، نرخ تکرار و کوتاهی‌های متوسط سیگنال، انرژی و توان سیگنال، باندهای طیفی، شتاب، سرعت تغییر و ... استخراج می‌شوند. هر یک از این ویژگی‌ها، نشان دهنده‌ی خصوصیات خاصی از سیگنال صوتی هستند و می‌توانند به نتایج بهتر در وظایف تشخیص و تفکیک صدا، ترجمه صوتی و دیگر وظایف پردازش را عامل شود.

در زیر فیچر های مطرح شده در این پروژه را مطرح مینماییم

MCFF .۱.۲.۱

MFCC یا مجموعه ویژگی‌های همینوسیکل فرکانسی، یک روش استخراج ویژگی از سیگنال صوتی است که معمولاً برای تشخیص و تمایز دادن بین صداها استفاده می‌شود. این روش بر اساس تجزیه سیگنال صوتی به بخش‌های مختلف با استفاده از نمونه‌های زمانی کوتاه تقسیم می‌شود. پس از تقسیم سیگنال، معیارهای آماری مانند انرژی، میانگین و واریانس برای هر بخش محاسبه شده و سپس با استفاده از تبدیل فوریه معیارهای زمانی به معیارهای فرکانسی تبدیل می‌شوند. در نهایت، ضرایب MFCC حاصل به عنوان بردار ویژگی برای تشخیص و تمایز دادن بین صداها استفاده می‌شوند.

Chorma_stft .۲.۲.۱

Chroma_STFT یک روش استخراج ویژگی از سیگنال صوتی است که برای تشخیص و تمایز دادن بین آکوردهای موسیقی استفاده می‌شود. این روش بر اساس تبدیل فوریه کوتاه مدت زمانی (STFT) سیگنال صوتی استخراج می‌شود. در این روش، ابتدا سیگنال صوتی تقسیم می‌شود و برای هر بخش کوتاه زمانی STFT محاسبه می‌شود. سپس با استفاده از STFT، انرژی هر باند فرکانسی در هر بخش محاسبه می‌شود. در نهایت، از انرژی هر باند فرکانسی برای محاسبه ماتریس کروما استفاده می‌شود.

ماتریس کروما نشان می‌دهد که در هر بخش زمانی، آکوردهای مختلف موسیقی در چه میزان وجود دارند. این ماتریس به طور معمول از ۱۲ ستون تشکیل شده است که هر ستون نمایانگر یک آکورد موسیقی

است. ارزش هر سلول در ماتریس نشان دهنده حضور یا عدم حضور آکورد مربوطه در بخش زمانی موردنظر است. در کل، Chroma_STFT یک روش مفید است که برای تحلیل و تمایز آکوردهای موسیقی در سیگنال صوتی استفاده می‌شود.

Spectral_Contrast .۳.۲.۱

Spectral_Contrastیک روش استخراج ویژگی از سیگنال صوتی است که برای تشخیص تفاوت‌های طیفی در آن استفاده می‌شود. این روش بر اساس تبدیل فوریه کوتاه مدت زمانی (STFT) سیگنال صوتی استخراج می‌شود.

در این روش، ابتدا سیگنال صوتی تقسیم می‌شود و برای هر بخش کوتاه زمانی STFT محاسبه می‌شود. سپس با استفاده ازSTFT ، طیف فرکانسی برای هر بخش محاسبه می‌شود. درSpectral_Contrast ، برای هر باند فرکانسی معیارهای مختلفی مانند میانگین و واریانس طیف در ناحیه آن باند محاسبه می‌شوند. سپس با استفاده از این معیارها، کنتراست طیفی بین باندهای مختلف محاسبه می‌شود.

مقادیر کنتراست طیفی نشان می‌دهد که در هر بخش زمانی، تفاوت‌های طیفی در باندهای مختلف برجسته است یا خیر. این ویژگی می‌تواند برای تشخیص و تمایز دادن بین سیگنال‌های صوتی با مشخصات طیفی متفاوت مانند صداهای موسیقی ، گفتار و غیره استفاده شود..

در کل، Spectral_Contrastیک روش مفید است که برای تشخیص ویژگی‌های طیفی در سیگنال صوتی و تمایز دادن بین آنها استفاده می‌شود.

Spectral_Centrolid .۴.۲.۱

در مورد استخراج ویژگی های داده های صوتی، مرکز طیفی (Spectral Centroid) یک ویژگی است که به طور کلی استفاده می شود. این ویژگی نقطه مرکز ثقل طیف سیگنال صدا را نشان می دهد. برای محاسبه آن، از میانگین وزن داده های فرکانسی موجود در سیگنال استفاده می شود، که وزن ها توسط بردارهای میدان طیفی مربوطه تعیین می شود. مرکز طیفی اطلاعاتی درباره توزیع محتوای فرکانسی یک سیگنال صدا ارائه می دهد. مقدار بالاتر مرکز طیفی نشان دهنده این است که بیشتر از انرژی سیگنال در فرکانس های بالاتر تمرکز شده است، در حالی که مقدار پایین تر نشان دهنده تمرکز بر فرکانس های پایین تر است. مرکز طیفی می تواند به عنوان یک ویژگی در برنامه های مختلف مانند تشخیص گفتار، طبقه بندی سبک موسیقی و طبقه بندی رویدادهای صوتی مورد استفاده قرار گیرد. این ویژگی می تواند برخی از ویژگی های طیفی یک صدا را نشان دهد و به تمایز بین انواع مختلف صداها بر اساس محتوای فرکانسی آنها کمک کند.

سه فیچر مختلف را میتوان از این داده استخراج نمود که میانگین وزن های فرکانسی ، ماکزیمم آن ها و همچنین مقدار میانه این داده بعنوان سه فیچر مجزا برای این تحلیل به کار گرفته شده است.

Zero_Crossing_rate .۵.۲.۱

Zero Crossing Rate (نرخ گذار صفر) یکی از ویژگی‌های استخراج شده از داده‌های صوتی است. این ویژگی نشان می‌دهد که چقدر سیگنال صوتی از صفر عبور می‌کند. وقتی که سیگنال صوتی از مثبت به منفی یا از منفی به مثبت تغییر می‌کند، یک گذار صفر رخ می‌دهد. نرخ گذار صفر می‌تواند اطلاعاتی درباره ریتم و تناوب سیگنال صوتی ارائه دهد. در سیگنال‌های با ریتم بالا، نرخ گذار صفر بیشتر می‌شود به این معنی که سیگنال صوتی بین مثبت و منفی بیشتر عوض می‌شود. در سیگنال‌هایی با ریتم کمتر، نرخ گذار صفر کمتر می‌شود زیرا سیگنال صوتی بین مثبت و منفی کمتر تغییر می‌کند. استفاده از نرخ گذار صفر در برنامه‌های پردازش صوتی می‌تواند در شناسایی الگوها، تشخیص گفتار، تشخیص خواننده یا سخنران و حتی تحلیل سیگنال‌های موسیقی مفید باشد.

Piptrack .۶.۲.۱

Piptrack یک ویژگی در پردازش و تحلیل گفتار است که برای تخمین فرکانس بنیادی (یا پیچیدگی یا تن طبیعی) یک سیگنال صوتی استفاده می‌شود. فرکانس بنیادی نشان دهنده نت مورد شنیدار و تن طبیعی یک صدا است. Piptrack بر پایه مفهوم تحلیل هارمونیک استوار است، جایی که اجزای هارمونیکی در طیف فرکانسی یک سیگنال صوتی شناسایی می‌شوند. این روش، قله‌های طیفی یا پیک‌های موجود در دامنه فرکانس را تحلیل کرده و حرکت آنها در طول زمان را پیگیری می‌کند. با پیگیری این قله‌ها، Piptrack فرکانس بنیادی را تخمین می‌زند و اطلاعاتی درباره تغییرات تن طبیعی در سیگنال صوتی ارائه می‌دهد. این ویژگی رایج در وظایفی مانند تشخیص گفتار، سنتز گفتار و تحلیل موسیقی استفاده می‌شود. این ویژگی به استخراج اطلاعات مربوط به فرکانس بنیادی، شناسایی الگوهای ملودیک و تمایز صداها یا سازهای مختلف بر اساس فرکانس بنیادی آنها کمک می‌کند.

Harmonic .۷.۲.۱

هارمونیک‌ها در تشخیص ویژگی‌های صدا در تحلیل گفتار نقش بسیار مهمی دارند. در گفتار، هارمونیک‌ها ضرب‌های صحیحی از فرکانس بنیادی (کمترین جزء فرکانسی) سیگنال صوتی هستند.

هارمونیک‌ها برای نشان دادن پوشش طیفی سیگنال صوتی استفاده می‌شوند و برای شناسایی تن (پیچیدگی) صدا بسیار حائز اهمیت هستند. حضور هارمونیک‌ها در سیگنال نشان می‌دهد که صدا دوره‌ای است، و عدم حضور هارمونیک‌ها نشان می‌دهد که صدا بی‌دوره یا نویزی است.

تشخیص هارمونیک‌ها در برنامه‌های مختلفی مانند تشخیص گفتار، تشخیص گوینده و تحلیل صدای خواننده استفاده می‌شود. این ویژگی در شناسایی فرکانس بنیادی، تعیین کیفیت صدا، و تشخیص حضور برباده‌های صدایی یا ناهنجاری‌های صدایی کمک می‌کند.

به طور خلاصه، تشخیص هارمونیک‌ها در تشخیص ویژگی‌های صدا بسیار حیاتی است زیرا اطلاعاتی درباره تن و پوشش طیفی سیگنال صوتی ارائه می‌دهند.

Energy .۸.۲.۱

اندازه گیری میزان انرژی یک سیگنال دارای اطلاعات زیادی میباشد از جمله تشخیص تن صدا برای بیان یک جمله ، همچنین قدرت همچنین صدا های با انرژی بیشتر قابلیت این را دارند که بتوانند راحتتر تشخیص داده شوند و تاثیر نویز در آن ها کمتر بوده و بعنوان یک فیچر میتوان از آن ها استفاده نمود.

Beat_track .۹.۲.۱

Beat trackingیا تشخیص ضرب در موسیقی یک ویژگی از تحلیل صدا است که به کمک آن می‌توان فاصله زمانی بین ضربات را در موسیقی تشخیص داد. این ویژگی معمولاً در تحلیل پردازشی موسیقی بکار می‌رود. در ویژگی های صدا، تشخیص ضرب در موسیقی با استفاده از الگوریتم های مختلفی انجام می‌شود که توسط تحلیل موجک، تبدیل فوریه، تحلیل انرژی و غیره انجام می‌شوند.

تشخیص ضرب در موسیقی در ویژگی های صدا، همچنین می‌تواند در تحلیل ویژگی های صدای انسانی نیز مفید باشد. به عنوان مثال، با تشخیص ضرب در موسیقی با استفاده از روش های مختلف، می توان به بررسی تغییرات فاصله‌های ضربات در زمان در صداهای انسانی پرداخت.

از دیگر کاربردهای تشخیص ضرب در موسیقی به عنوان یک قابلیت صدای هوشمند در تلفن همراه و دستگاه های پخش موسیقی می توان اشاره کرد. با افزودن کاربرد تشخیص ضرب در موسیقی به این دستگاه ها، می توان به طور خودکار تنظیم های مختلف را برای پخش موسیقی انجام داد. به طور مثال، با تشخیص ضرب در موسیقی، دستگاه پخش موسیقی می تواند به طور خودکار حالت های مختلفی را اعم از پخش در حالت شلوغی یا آرامش و... به طور خودکار انتخاب کند.

بخش ۲. طبقه‌بندی داده ها

در بخش قبل مراحل تمیز کردن دیتا (Data cleaning) و همچنین استخراج ویژگی از روی داده های صوتی بیان شد. در این بخش قصد داریم تا از ویژگی هایی که در بخش قبل استخراج شد، برای طبقه بندی داده داده های صوتی در دو طبقه "Male" و "Female" استفاده کنیم.

طبقه بندی نوعی از Binary Classification است که راه های مختلفی برای جداسازی داده های مرد و زن می توان استفاده کرد. از بهترین آنها می توان به روش های زیر اشاره کرد:

- SVM
- Random Forest
- KNN
- Decision Tree

۱.۲. مدل های پیشنهادی

در دانشگاه VIT پژوهش انجام شد برای طبقه بندی داده های صوتی زن و مرد. در این مقاله از چندین روش برای طبقه بندی استفاده شد که دو تا از بهترین روش هایی که گزارش داده است، SVM و Random Forest است. در واقع SVM بیشتر دقت و RF بیشترین مقاومت (Robustness) را در میان سایر روش ها داشته است[1].

۲.۲. روش های بهبود نتیجه

در درس، روش هایی برای بهبود نتایج طبقه بندی تدریس شد مانند کاهش بعد (Dimensionality reduction) که یک ابزار ارزشمند برای بهبود عملکرد مدل است که از ویژگی های به نسبت کم ارزش چشم‌پوشی می کند و همچنین نرمالایز کردن(Normalization) یا اسکیل کردن فیچرها که یکی از مراحل مهم پیش پردازش است و برای مدل هایی که به اندازه فیچرهایی ورودی حساس هستند حیاتی است. در زیر توضیح مختصری در مورد نحوه عملکرد روش های بهبود نتیجه‌ی ذکر شده داده می شود.

- کاهش بعد: از PCA برای کاهش بعد استفاده شده است. بدین ترتیب که تعداد فیچر ها به ۲۰ فیچر کاهش یافته است.
- کاهش بعد: علاوه بر روش قبلی، می توان از تکنیک LDA (Linear Discremenant analysis) نیز استفاده کرد.

- نرمالایز کردن: با استفاده از کتابخانه StandardScaler، مقادیر فیچر ها نرمالایز شده اند تا در ادامه بتوان از آن ها برای یادگیری مدل SVM و RF استفاده کرد.

فایل transcripts_features.csv توسط کتابخانه pandas خوانده می شود. ۵ ردیف اول و همچنین توصیفات این فایل در زیر قابل مشاهده است.

داده ها به دو دسته train و test و با نسبت ۷۵٪-۲۵٪ تقسیم می شوند طوری که هیچکدام از این دو دسته نسبت به داده های male و female بایاس نباشند. در واقع چون تعداد داده های male بیشتر است امکان دارد این اتفاق بیافتد.

```
Training Set Class Counts:
gender
male      3540
female     991
Name: count, dtype: int64

Testing Set Class Counts:
gender
male      1180
female     331
Name: count, dtype: int64
```

شکل ۱. تعداد نمونه های مرد و زن در داده‌های آموزشی و تست

```
X_train.describe()
```

	Unnamed: 0	centroid_mean	bandwidth_mean	zero_crossings_mean
count	4531.000000	4531.000000	4531.000000	4531.000000
mean	3007.928934	2952.598856	1834.282469	0.293119
std	1740.374556	506.671377	237.001735	0.063177
min	0.000000	1211.383990	964.516766	0.111397
25%	1511.000000	2627.908619	1709.002165	0.250791
50%	2999.000000	2951.787423	1888.433900	0.284359
75%	4510.500000	3298.698891	2000.063184	0.330043
max	6041.000000	4487.562284	2461.088524	0.554474

شکل ۲. توصیف داده‌های آموزشی

در ادامه، فیچرهایی که شامل چند مقدار بوده اند بسط داده شده اند. به طور مثال برای ستون mfcc_mean که شامل یک لیست با ۱۳ مقدار است، ۱۳ ستون مجزا در نظر گرفته شده است تا بتوان در ادامه آن ها را اسکیل کرد.

```
print("X_train shape before expanding the columns")
X_train.shape
```

```
X_train shape before expanding the columns
(4531, 18)
```

شکل ۳. ابعاد داده‌ی آموزشی قبل از بسط دادن

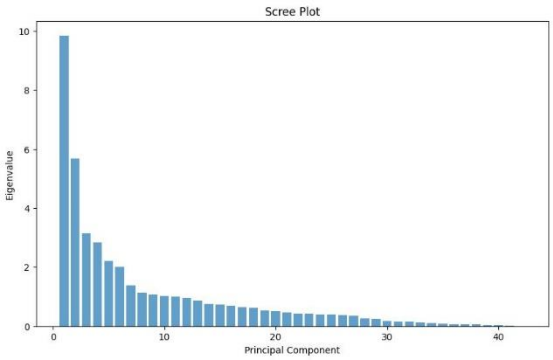
```
print("X_train shape after expanding the columns")
X_train.shape
```

```
X_train shape after expanding the columns
(4531, 47)
```

شکل ۴. ابعاد داده‌ی آموزشی پس از بسط دادن

همانطور که مشاهده می شود، ابعاد هر نمونه آموزشی زیاد است. بنابراین به کمک الگوریتم های کاهش بعد یعنی **PCA** و **LDA** اقدام به کاهش شده و در ادامه نتایج هر الگوریتم طبقه بندی برای داده‌های کاهش بعد یافته گزارش می شود.

برای یافتن بهترین تعداد component در الگوریتم PCA، به دنبال بیشینه کردن واریانس هستیم. برای بیشینه کردن واریانس، بردارهایی مفید هستند که مقدار ویژه متناظر با آن ها بیشینه باشند. نمودار مقادیر ویژه را به ازای component ها رسم می شود.



شکل ۵. نمودار مقادیر ویژه بر حسب **component**ها

نمودار در n=8 شکستگی دارد بنابراین از همین عدد به عنوان آرگومان ورودی الگوریتم PCA استفاده می کنیم.

```
X_train_pca.shape
```

```
(4531, 8)
```

شکل ۶. ابعاد داده ها پس از اعمال الگوریتم **PCA**

همچنین از LDA نیز برای کاهش بعد داده ها استفاده شده است.

```
X_train_lda.shape
```

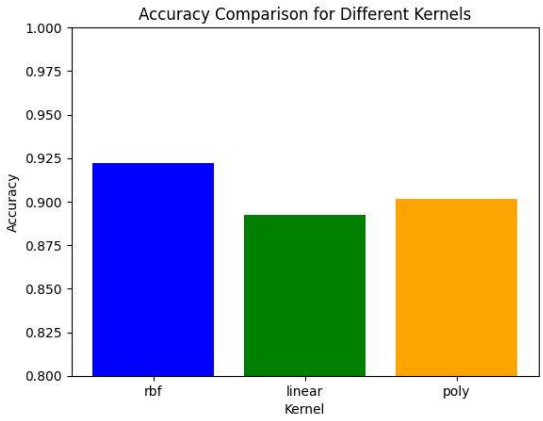
```
(4531, 1)
```

شکل ۷. ابعاد داده ها پس از اعمال الگوریتم **LDA**

۱. طبقه بندی با الگوریتم SVM

مدل اول برای طبقه بندی صداهای مرد و زن توسط الگوریتم SVM(Support Vector Machine) آموزش داده شد. نتیجه دقت آموزش این مدل برای سه کرنل مختلف یعنی RBF، خطی و چند جمله ای در زیر قابل مشاهده است.

در این بخش ابعاد ویژگی از ۴۰ تا با کمک PCA به ۸ فیچر کاهش داده شده است.



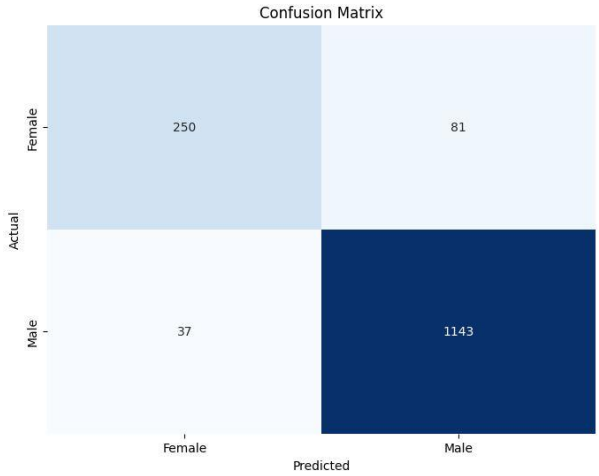
شکل ۸. دقت آموزش مدل **SVM** توسط ۳ کرنل مختلف

مشاهده می شود که کرنل **rbf** بیشترین دقت یعنی ۹۲٪ را داشته است که مقادیر **recall**، **f1-score** و صحت آن در شکل زیر نیز قابل مشاهده است.

جدول ۱ نتایج کرنل **rbf**

```
Classification Report:
              precision    recall  f1-score   support

      female       0.87      0.76      0.81       331
        male       0.93      0.97      0.95      1180

    accuracy                           0.92      1511
   macro avg       0.90      0.86      0.88      1511
weighted avg       0.92      0.92      0.92      1511

Accuracy: 0.9219060225016545
```

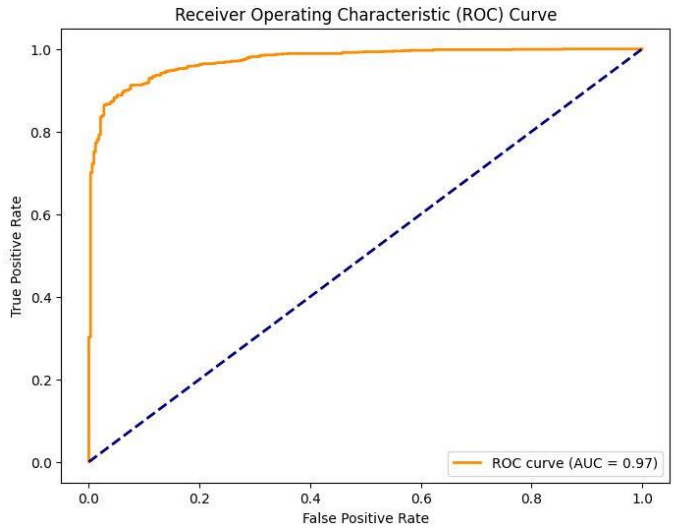
ماتریس آشفتگی مدل SVM با کرنل rbf در شکل زیر قابل مشاهده است.



شکل ۹. ماتریس آشفتگی کرنل **rbf**

ماتریس آشفتگی دقت بسیار بالای طبقه بندی صدای مرد را گزارش می دهد که از ۱۱۸۰ نمونه تنها ۳۷ عدد از آنها را در دسته‌ی صدای زن قرار داده است. اما در طبقه بندی صدای زن، در ۳۳۱ نمونه، ۸۱ مورد را صدای مرد تشخیص داده است.

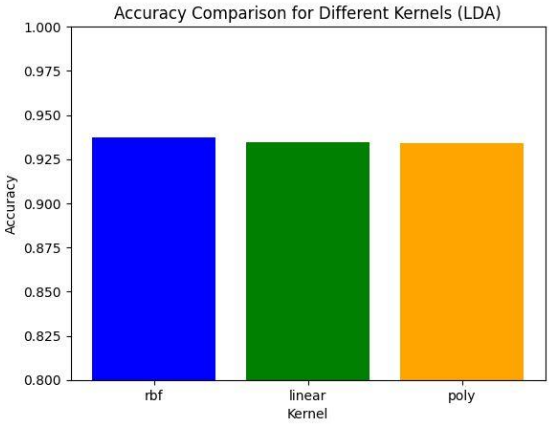
نمودار ROC در تصویر زیر قابل مشاهده است.



شکل ۱۰. نمودار **ROC**

سطح زیر نمودار ROC نزدیک حدود ۰.۹۲ است که نشان از عملکرد نسبتا خوب مدل دارد.

در ادامه، ابعاد فیچرها به کمک تکنیک LDA کاهش داده شده و سپس نتایج ارائه می شوند. تعداد component ها در تکنیک LDA، باید از min(features, classes-1) کمتر باشد بنابراین در این بخش آرگومان n_components=None در نظر گرفته شده است.



شکل ۱۱. دقت آموزش مدل **SVM** توسط ۳ کرنل مختلف (**LDA**)

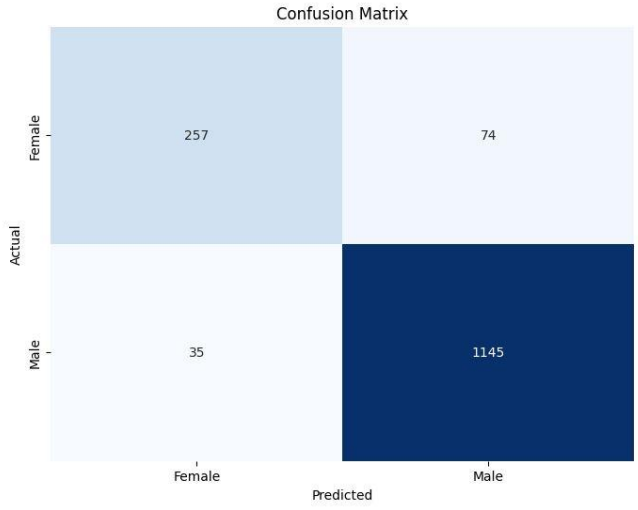
در اینجا نیز مانند حالت قبل، کرنل rbf بهترین عملکرد را داشته است که دقت و ماتریس آشفتگی آن در شکل زیر قابل مشاهده است.

Classification Report:

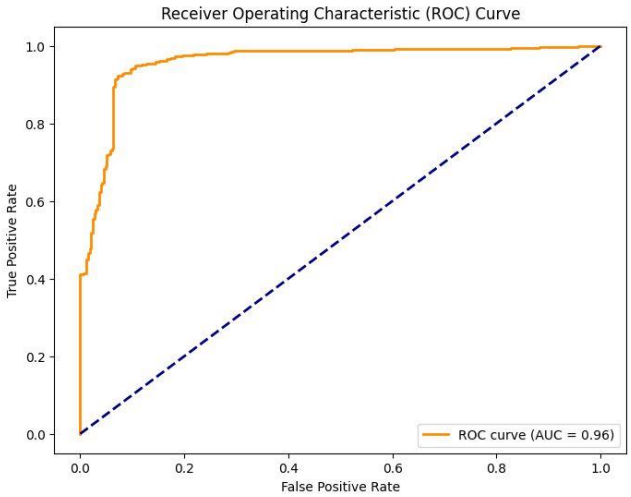
	precision	recall	f1-score	support
female	0.88	0.82	0.85	331
male	0.95	0.97	0.96	1180
accuracy			0.94	1511
macro avg	0.92	0.90	0.91	1511
weighted avg	0.94	0.94	0.94	1511

Accuracy: 0.9371277299801456

شکل ۱۲. نتایج کرنل **rbf** (**LDA**)



شکل ۱۳. ماتریس آشفتگی کرنل **rbf** (**LDA**)



شکل ۱۴. نمودار **ROC** (**LDA**)

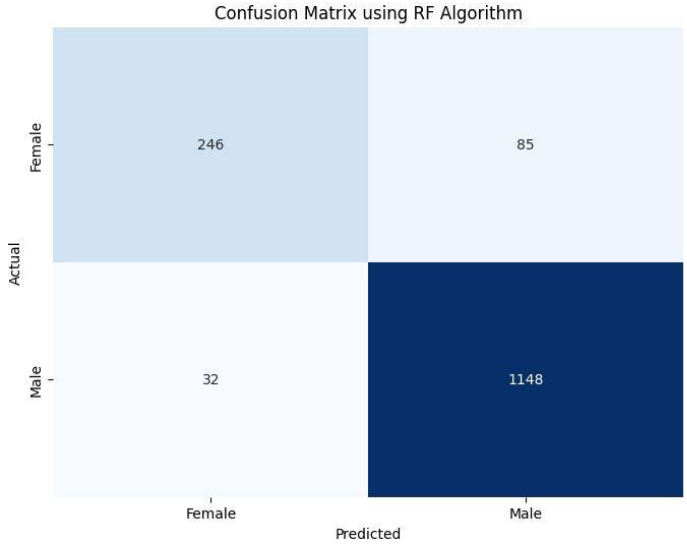
۲. طبقه بندی با الگوریتم Random Forest

در این بخش به طبقه بندی با استفاده از الگوریتم RF می پردازیم.

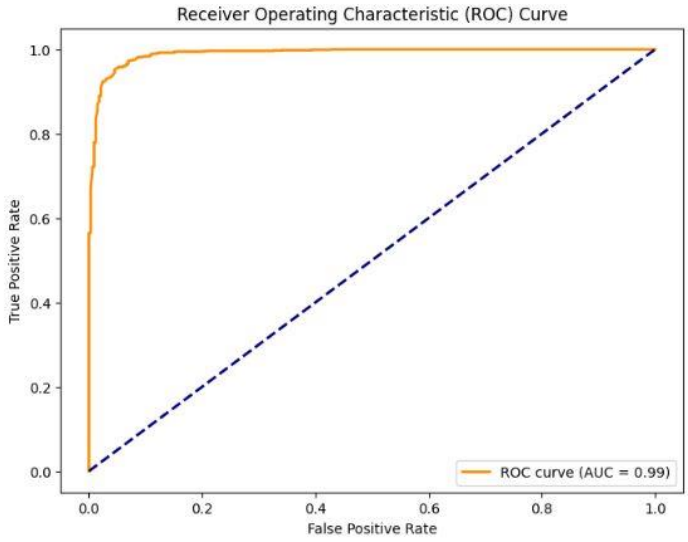
نتایج در ادامه قابل مشاهده است. در نتایج زیر، از PCA برای کاهش ابعاد فیچر ها استفاده شده است.

```
Classification Report:
              precision    recall  f1-score   support

      female       0.88      0.74      0.81       331
        male       0.93      0.97      0.95      1180

    accuracy                           0.92      1511
   macro avg       0.91      0.86      0.88      1511
weighted avg       0.92      0.92      0.92      1511

Accuracy: 0.9225678358702846
```

شکل ۱۵. نتایج طبقه بندی با الگوریتم **RF** با دقت نهایی **92.25%**



شکل ۱۶. ماتریس آشفتگی الگوریتم **RF**



شکل ۱۷. نمودار **ROC** الگوریتم **RF**

۳. طبقه بندی با الگوریتم Logistic Regression

در این بخش به طبقه بندی با استفاده از الگوریتم Logistic Regression می پردازیم.

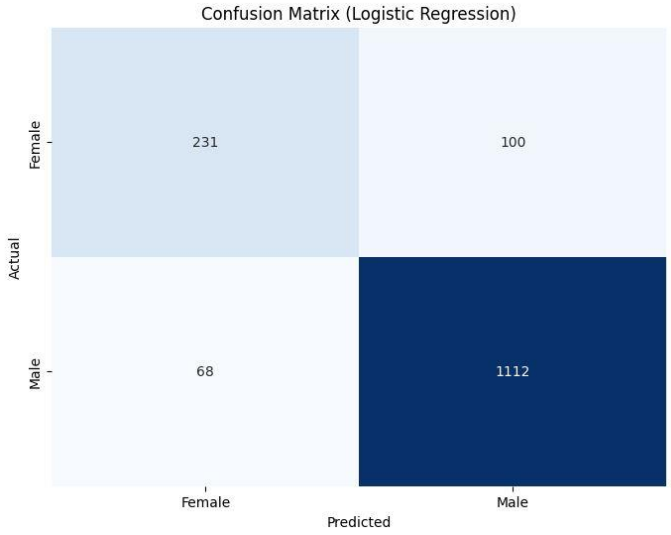
در این بخش نیز از داده هایی استفاده می شود که توسط الگوریتم PCA کاهش بعد داده شده اند.

نتایج در ادامه قابل مشاهده است.

```
Classification Report for logistic regression:
              precision    recall  f1-score   support

      female       0.88      0.74      0.81       331
        male       0.93      0.97      0.95      1180

    accuracy                           0.92      1511
   macro avg       0.91      0.86      0.88      1511
weighted avg       0.92      0.92      0.92      1511

Accuracy: 0.8888153540701522
```

شکل ۱۸. نتایج طبقه بندی با الگوریتم **Logistic regression** با دقت نهایی **88.88**٪

در صورتی که از داده های اصلی (داده هایی که بعد آنها کاهش داده نشده اند) استفاده میشد، دقت حدود ۹۲٪ است.

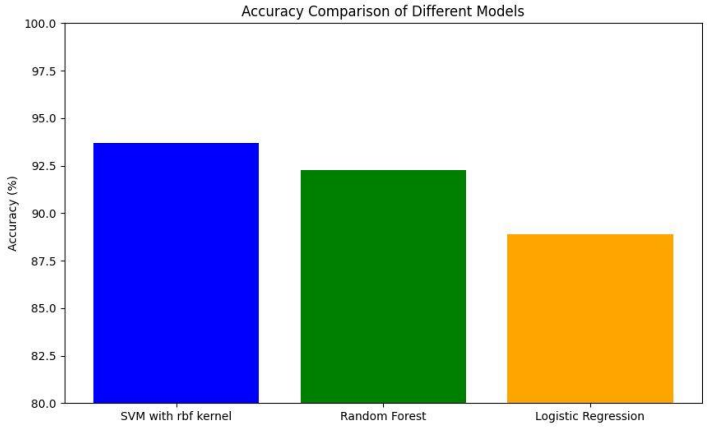


شکل ۱۹. ماتریس آشفتگی الگوریتم **Logistic Regression**

۳.۲. نتیجه‌گیری و مقایسه

در این بخش از ۳ الگوریتم مختلف برای طبقه بندی صوت استفاده شده است.

در تصویر زیر مقایسه دقت این سه روش قابل مشاهده است.



شکل ۲۰. مقایسه سه الگوریتم استفاده شده برای طبقه بندی صدای مرد و زن

همانطور که در شکل بالا مشخص است، هر سه مدل دقت خوبی داشته اند اما دقت الگوریتم SVM با کرنل rbf کمی دقت بالاتری(۹۳٪) داشته است.

همچنین از ماتریس های آشفتگی مشخص است که هر سه الگوریتم در طبقه بندی صدای Female اشتباه بیشتری در طبقه بندی داشته اند. یکی از دلایل آن می تواند ناشی از بیشتر بودن تعداد داده ی صوتی Male باشد. اگرچه با تقسیم مناسب داده ها در داده های آموزشی و تست، سعی شد از بایاس شدن نتیجه جلوگیری شود. تعداد داده های اصلی male، حدود 3.5 برابر تعداد داده های female بود. با این حال هر ۳ الگوریتم عملکرد مناسبی داشتند.

همچنین استفاده از الگوریتم های کاهش بعد مانند LDA و PCA منجر به کاهش دقت شد چون از تمامی ویژگی های صوتی استفاده نشده است. اما با این حال هزینه محاسباتی و همچنین زمان یادگیری مدل را کاهش داده است.

بخش ۳. خوشه بندی داده ها

۱.۳. مدل های پیشنهادی

برای خوشه‌بندی داده‌ها، مدل‌های مختلفی وجود دارند. مانند:

۱- خوشه‌بندی سلسله مراتبی (Hierarchical Clustering):

این مدل به ما این امکان را می‌دهد که به صورت سلسله مراتب ، خوشه‌ها را بسازیم. این امر می‌تواند مفید باشد اگر بخواهیم ساختار دقیق‌تری از داده‌هایمان را بدست آوریم.

۲-:DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

این الگوریتم بر اساس چگالی نقاط به جای تعداد خوشه‌ها عمل می‌کند. این الگوریتم می‌تواند با وجود داده‌های نویزی و اندازه‌های متفاوت خوشه‌ها مقاوم‌تر باشد

۳- الگوریتم K-Means :

در انجام این پروژه از روش الگوریتم K-Means استفاده کرده ایم.

الگوریتم K-Means با انجام مراحل مقداردهی اولیه و به‌روزرسانی تا همگرایی، داده‌ها را به خوشه‌های مختلف تقسیم کرده و مراکز خوشه‌ها را به موقعیت بهینه رسانده است.

با استفاده از روش‌های بهبودی مانند کاهش بعد و نرمال‌سازی، کیفیت و عملکرد مدل بهبود یافته است.

۲.۳. روش های بهبود نتیجه

۱-کاهش بعد (Dimensionality Reduction) :

استفاده از روش‌های کاهش بعد مانند PCA (Principal Component Analysis) یا t-SNE می‌تواند کمک کند تا داده‌های پیچیده‌تر را در یک فضای کمتر ابعاد قرار دهیم و الگوریتم خوشه‌بندی بهبود یابد.

۲- نرمال‌سازی(Normalization) :

نرمال‌سازی ویژگی‌ها می‌تواند به ازدیاد کارایی الگوریتم‌های خوشه‌بندی کمک کند. این عمل باعث می‌شود تا واحد مقیاس ویژگی‌ها یکسان شود و الگوریتم بهتر بتواند تطابق مناسبی انجام دهد.

۳- انتخاب تعداد خوشه مناسب:

استفاده از روش‌هایی مانند نمودار پراکندگی خوشه‌ها و ارزیابی معیارهایی نظیر silhouette score برای انتخاب بهترین تعداد خوشه ممکن است به بهبود نتایج بیانجامد.

۴- استفاده از معیارهای ارزیابی:

معیارهای ارزیابی مانند دقت (accuracy) یا F1-score می‌توانند کمک کنند تا عملکرد مدل‌های خوشه‌بندی را بهبود بخشید.

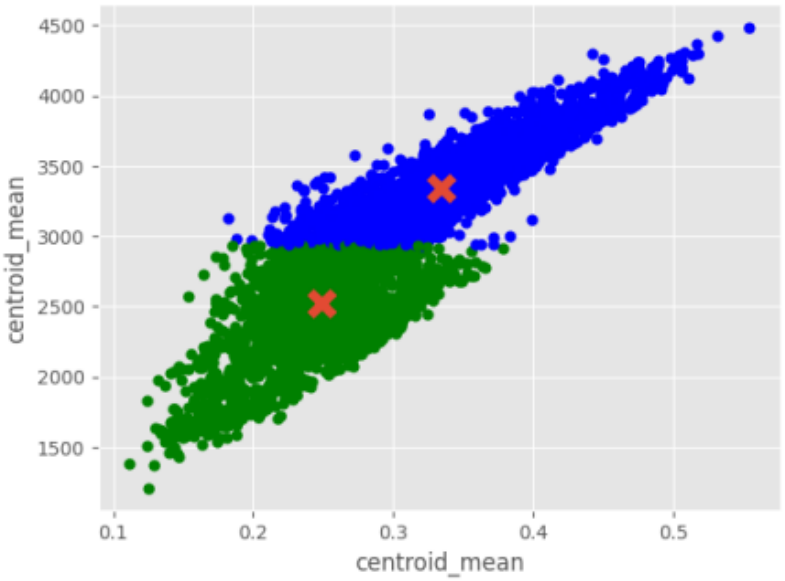
۵- استفاده از روش‌هایی مانند نمودار پراکندگی خوشه‌ها و ارزیابی معیارهایی نظیر silhouette score برای انتخاب بهترین تعداد خوشه ممکن است به بهبود نتایج بیانجامد.

با اعمال این روش‌ها، می‌توانید بهبودهای چشمگیری در کارایی مدل‌های خوشه‌بندی خود داشته باشید.

در اینجا ما از خوشه‌بندی k-means با تعداد خوشه‌های مختلف (۲و ۳ و ۴) با استفاده از ویژگی‌های 'zero_crossings_mean'و 'centroid_mean' از مجموعه داده هایمان استفاده کرده‌ایم. هر خوشه با رنگهای مختلف هستند و برای هر خوشه نشانگر 'x' برای مراکز خوشه‌ها نمایش داده شده است.

۳.۳. نتایج هر حالت:

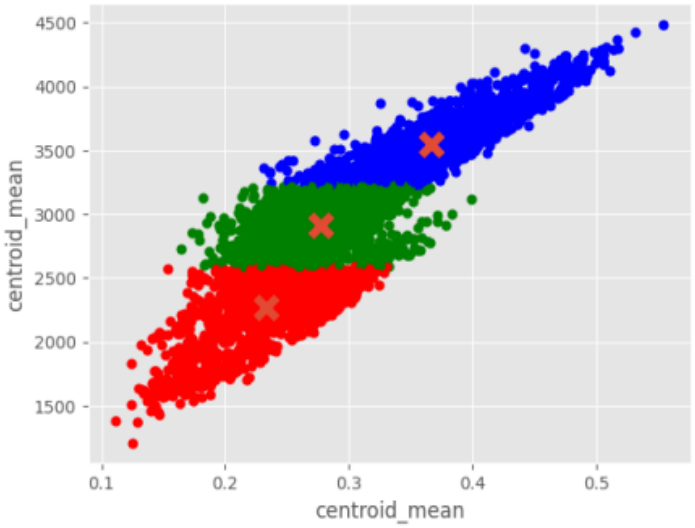
حالت ۲خوشه:



داده به دو خوشه تقسیم شده است، نقاط سبز یک خوشه و نقاط آبی خوشه دیگر را نمایان می‌کنند.

به نظر می‌آید که خوشه‌ها تفکیک واضحی دارند و مراکز با 'x' مشخص شده‌اند.

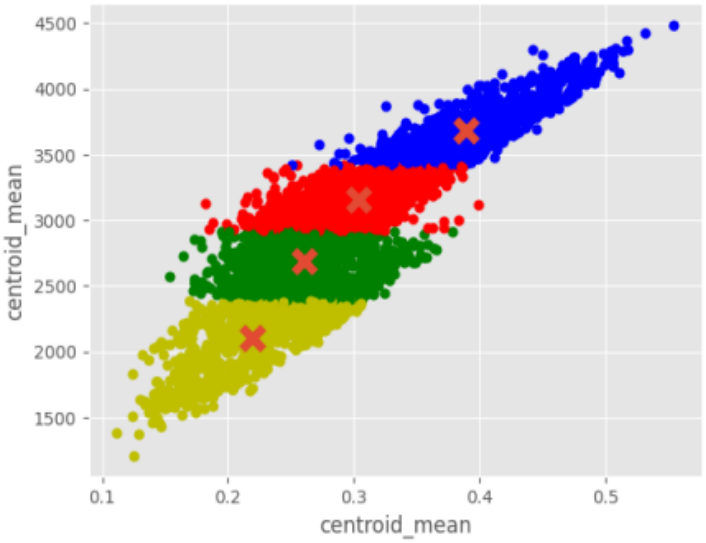
حالت ۳خوشه:



داده حالا به سه خوشه تقسیم شده است، با نقاط سبز، آبی و قرمز به ترتیب هر خوشه را نمایان می‌کنند.

افزودن یک خوشه اضافه به دقت بیشتر در خوشه‌بندی منجر شده و الگوریتم به نحوی الگوهای دقیق‌تری از داده را ضبط کرده است.

حالت ۴خوشه:



داده به چهار خوشه تقسیم شده است، با نقاط سبز، آبی، قرمز و زرد به ترتیب هر خوشه را نمایان می‌کنند.

با اضافه کردن یک خوشه دیگر، دقت بیشتری در تجزیه و تحلیل داده‌ها حاصل شده و هر خوشه یک زیرمجموعه خاص از نقاط را نمایان می‌کند.

۳.۳. نتیجه گیری و مقایسه

نتیجه‌گیری و مقایسه مدل‌های خوشه‌بندی با استفاده از الگوریتمK-Means :

۱- میزان پراکندگی:

میزان پراکندگی درون خوشه (Intra-Cluster Dispersion) : این معیار نشان دهنده این است که داده‌های درون هر خوشه به چه اندازه نزدیک به یکدیگر هستند.

۲- میانگین پراکندگی میان خوشه (Inter-Cluster Dispersion) : این معیار نشانگر این است که چقدر خوشه‌ها از یکدیگر دور هستند.

۳- تحلیل ویژگی‌های هر خوشه:

برای هر تعداد خوشه، نماینده بودن یک خوشه از ویژگی‌های مشترک داده‌های درون آن خوشه نتیجه می‌شود.

به عنوان مثال، اگر دو خوشه تشخیص داده شوند(K=2) ، ممکن است یک خوشه نماینده داده‌های مرتبط با ویژگی‌های صوتی مردان باشد، و دیگری نماینده داده‌های مرتبط با ویژگی‌های صوتی زنان.

۴- تعداد خوشه‌ها:

برای هر تعداد خوشه انتخابی، میزان شباهت داده‌های درون یک خوشه و تفاوت بین خوشه‌ها بررسی می شود.

ممکن است برخی داده‌ها در یک خوشه به دلیل شباهت در ویژگی‌ها گروه‌بندی شده باشند. تفاوت‌های بین خوشه‌ها ممکن است به تفکیک صحیح میان دسته‌ها کمک کند.

مزیت های استفاده از الگوریتم K-Means :

۱- سادگی و سرعت بالا:

K-Meansیک الگوریتم سریع و ساده است که به راحتی قابل استفاده و تفسیر است.

۲- کارایی در مقیاس‌های بزرگ:

برای مجموعه داده‌های بزرگ، K-Means به خوبی مقیاس می‌شود و به سرعت می‌تواند خوشه‌بندی را انجام دهد.

۳- قابلیت مشخص کردن تعداد خوشه‌ها:

قابلیت انتخاب تعداد خوشه‌ها توسط کاربر یک امکان مهم است.

۴- استفاده از K-Means با توجه به سادگی، سرعت، و قابلیت مشخص کردن تعداد خوشه‌ها، مزایای زیادی دارد. با توجه به مطالب بالا، این الگوریتم می‌تواند در مسائل مختلفی که نیاز به خوشه‌بندی دارند، مفید باشد.

در این پروژه با به کارگیری الگوریتم K-Means و با افزایش تعداد خوشه ها از دو به سه و سپس چهارتا خوشه، دقت طبقه بندی افزایش و همچنین میزان پراکندگی درون خوشه کاهش یافت.

بخش ۴. مدل Automatic Speech Recognition (ASR)

۱.۴. مقدمه

نام ASR به تکنولوژی ای گفته می شود که گفتار را به متن تبدیل می کند. به طور کلی دو دسته روش برای انجام این کار وجود دارد.

۱- روش های مختلف سنتی مانند روش های آماری (statistical)

۲- روش های مدرن مانند انتها به انتها (End-to-End) وجود دارد.

در این بخش به توضیح مختصری از این دو روش می پردازیم.

۱. روش سنتی – آماری

رویکردهای آماری سنتی اغل بر مدل های احتمالاتی و متد های آماری برای مدل سازی روابط بین ویژگی های صوتی و واحد های زبانی متکی هستند. برخی از روش های مرسوم و کلیدی عبارتند از:

۱- مدل های Hidden Markov

این مدل ها آماری هستند که دنباله ای از حالت های قابل مشاهده (ویژگی های صوتی) را از طریق یک سری حالت های پنهان (واحدهای آوایی یا واحد های شبه کلمه ای) نشان می دهد. انتقال بین این لایه ها به صورت احتمالاتی مدل می شوند.

۲- GMM ها

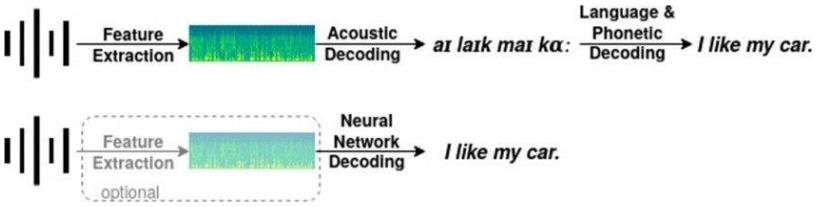
این مدل ها، توزیع احتمال را به صورت مجموع چند توزیع گوسی مدل می کند. در ASR، از این مدل ها برای مدل سازی توزیع ویژگی های صوتی مرتبط با واحدهای آوایی مختلف استفاده می شود.

۳- Vector Quantization

VQ یک تکنیک کوانتزیزه کردن است که برای نمایش بردارهایی با مقادیر پیوسته با مجموعه ای محدود از نمادهای گسسته به کار می رود. در ASR از این روش برای کوانتیزه کردن بردارهای ویژگی صوتی به مجموعه کوچکتری از بردار ها استفاده می شود. از این روش برای خوشه بندی feature space و رویکردهای مبتنی بر codebook برای کاهش ابعاد بردارهای ویژگی صوتی استفاده می شود.

۲. روش های مدرن – انتها به انتها

رویکرد انتها به انتها از روش های یادگیری عمیق (معمولا شبکه های عصبی) برای تبدیل کردن ویژگی های صوتی به متن و به صورت مستقیم استفاده می کند. در زیر تصویر از این pipeline قابل مشاهده است.



بطور کلی این روش از متدهای یادگیری عمیق بهره میگیرد تا سیگنال صوتی خام را مستقیما به متن تبدیل کند.

مقایسه:

- از نظر پردازش:

روش های انتها به انتها معماری ساده تری دارند و همچنین تعداد مراحل کمتری را نیز نیاز دارند اما محاسبات پیچیده ای دارند. زیرا ترین کردن مدل شبکه های عصبی ممکن است از لحاظ پردازی سنگین و وقت گیر باشند به خصوص اگر از مدل های بزرگ استفاده شود.

از طرفی روش های سنتی مانند GMM ها به علت قدیمی تر بودن، الگوریتم های بهنیه تری دارند و همچنین در مرحله decoding بازدهی بالاتری دارند. اما از طرف دیگر تعداد مراحل بیشتری برای تولید متن خروجی دارند.

- از نظر دقت:

روش های مدرن انتها به انتها به علت استفاده از شبکه های عمیق می تواند دقت بیشتری را فراهم کنند اما از طرفی این اتفاق زمانی می افتد که به اندازه کافی داده ورودی موجود باشد. اگر حجم داده زیاد بوده و محدودیت پردازشی وجود نداشته باشد، روش انتها به انتها می تواند دقت بسیار خوبی را فراهم کند.

- از نظر داده:

روش های انتها به انتها مبتنی بر شبکه عصبی عمیق است کع ورودی را به خروجی به صورت مستقیم مپ می کند. این روش تکیه زیادی بر حجم داده دارد اما نکته قابل توجه این است که این داده ها می توانند بسیار متنوع باشد. اما روش های مبتنی بر آمار، نیازمند حجم دیتای برچسب خورده هستند اما به نسبت با داده های کمتری می توانند عملکرد خود را داشته باشند همچنین حساسیت زیادی نسبت به کیفیت داده ها دارند.به طور کلی انتخاب میان این روش بستگی به تنوع داده، کیفیت داده و محدودیت توان پردازشی دارد.

۲.۴. مدل Wav2Vec به همراه Transformer

تشخیص خودکار گفتار (ASR) یک فناوری متحول کننده است که زبان گفتاری را به متن نوشتاری تبدیل می کند. سیستم‌های ASR کاربردهایی را در خدمات ویراستاری، دستیارهای صوتی و حوزه‌های مختلف پیدا می‌کنند که تبدیل گفتار به متن ضروری است. مدل ارائه شده در این گزارش بر اساس معماری Wav2Vec2، به ویژه مدل «Wav2Vec2ForCTC» است که از مدل پیش آموزش داده شده «facebook/wav2vec2-large-xlsr-53» استفاده شده است.

۱.۲.۴ نمای کلی مدل

مدل Wav2Vec2ForCTC

انتخاب مدل «Wav2Vec2ForCTC» برای کار ASR باتوجه به معماری قوی و عملکرد برتر آن در وظایف مربوط به گفتار پشتیبانی میباشد. این مدل بخشی از کتابخانه Hugging Face Transformers است و گونه‌ای از معماری Wav2Vec2 است که به‌طور خاص برای وظایف طبقه‌بندی زمانی ارتباطی[۱] (CTC) طراحی شده است که به خوبی با ماهیت ASR همسو می‌شود.

مدل "facebook/wav2vec2-large-xlsr-53" از پیش آموزش داده شده

این مدل با وزنه های از پیش آموزش داده شده بر روی یک دیتاست بزرگ، به طور خاص "wav2vec2-large-xlsr-53/facebook" لود شده است. این مدل از پیش آموزش‌دیده در طول مرحله پیش‌آموزشی، در معرض حجم وسیعی از داده‌های گفتاری چندزبانه و متنوع قرار گرفته است، و آن را قادر می‌سازد تا بازنمایی‌های پیچیده ویژگی‌های آوایی و آکوستیک را بیاموزد. وزن‌های از پیش آموزش‌دیده شده به‌عنوان یک مقدار اولیه ارزشمند عمل می‌کنند و به مدل اجازه می‌دهند تا دانش را از مفهوم وسیع‌تر درک گفتار به‌کار گیرد.

۲.۲.۴ معماری Wav2Vec2

استخراج ویژگی

لایه های اولیه مدل بر استخراج ویژگی از شکل موج های صوتی خام تمرکز دارد. مجموعه‌ای از لایه‌های کانولوشن، صدای ورودی را پردازش می‌کند و ویژگی‌های آکوستیک مربوطه مانند گام، شدت و ویژگی‌های طیفی را استخراج می‌کند. این مرحله در تبدیل سیگنال صوتی پیوسته به فرمتی مناسب برای تجزیه و تحلیل بعدی بسیار مهم است.

---

[۱] Connectionist Temporal Classification

Context Aggregation with Self-Attention

Wav2Vec2 از مکانیسم های خودتوجهی مبتنی بر ترانسفورماتور ها برای استخراج وابستگی دنباله های صدای ورودی استفاده می‌کند. این به مدل اجازه می‌دهد تا وابستگی‌های دوربرد را در نظر بگیرد و زمینه کلی را در دنباله صوتی ثبت کند. توجه به خود مدل‌سازی روابط بین بخش‌های مختلف توالی ورودی را تسهیل می‌کند و به توانایی مدل در تشخیص الگوهای ظریف در گفتار کمک می‌کند.

اتلاف CTC برای یادگیری انتها به انتها

این مدل از (CTC) در طول آموزش استفاده می‌کند. CTC مخصوصاً برای یادگیری انتها به انتها در ASR مناسب است، زیرا به تراز صریح بین فریم های صوتی ورودی و رونویسی های خروجی نیاز ندارد. درعوض، به مدل اجازه می‌دهد تا چنین هم‌ترازی‌هایی را به طور ضمنی یاد بگیرد و آن را با سرعت‌ها و الگوهای گفتار متفاوت سازگار کند.

Dropout برای استحکام

برای افزایش استحکام مدل، Dropout اعمال می شود. Hidden dropout (Dropout در لایه های پنهان) و Attention dropout (Dropout در لایه های توجه) به جلوگیری از برازش بیش از حد در طول آموزش کمک می کند. Layer-wise dropout، تنوع را در لایه‌های مختلف معرفی می‌کند، تعمیم را ایجاد می‌کند و خطر وابستگی بیش از حد مدل به اجزای خاص را کاهش می‌دهد.

Gradient Checkpointing

این مدل شامل نقطه کنترل گرادیان است، تکنیکی که به مدیریت مصرف حافظه در طول آموزش کمک می کند. با معاوضه کردن شدت محاسباتی برای کارایی حافظه، بررسی گرادیان امکان آموزش مدل‌های بزرگ‌تر را بر روی پردازنده‌های گرافیکی با ظرفیت حافظه محدود می‌دهد. این به ویژه در مورد معماری های عمیق مانند Wav2Vec2 مفید است.

۳.۲.۴ پیکربندی مدل

پارامترهای قابل تنظیم

انعطاف‌پذیری مدل از طریق هایپرپارامتر ها با دقت انتخاب شده افزایش می‌یابد:

I. Attention Dropout, Hidden Dropout, and Feat Proj Dropout: این پارامترها نرخ ترک تحصیل را کنترل می کنند، نظم بخشی و ظرفیت یادگیری را متعادل می کنند.
II. Mask Time Probability: معرفی پوشش زمانی در حین تمرین به قرار دادن مدل در معرض توالی های ورودی متنوع و تا حدی مبهم و افزایش استحکام کمک می کند.

III. LayerDrop: مشخص کردن احتمال افتادن یک لایه ترانسفورماتور در حین آموزش، امکان حذف کنترل شده از نظر لایه را فراهم می کند.

IV. CTC Loss Reduction: روش کاهش "میانگین" برای از دست دادن CTC در طول تمرین انتخاب می شود.

V. Pad Token ID and Vocabulary Size: پیکربندی مربوط به فرآیند توکن سازی و واژگان مورد استفاده مدل.

۴.۲.۴ آموزش و تنظیم دقیق

مدل تحت یک فرآیند آموزشی دو مرحله ای قرار می گیرد. ابتدا، روی یک مجموعه داده بزرگ و متنوع برای یادگیری بازنمایی های کلی گفتار از قبل آموزش داده شده است. متعاقباً، تنظیم دقیق مجموعه داده‌های ویژه کار، مدل را با تفاوت‌های ظریف و ویژگی‌های وظیفه ASR هدف هماهنگ می‌کند. این رویکرد آموزشی دو مرحله‌ای از مزایای پیش‌آموزش بدون نظارت و تنظیم دقیق نظارت‌شده استفاده می‌کند، که منجر به مدلی می‌شود که تعمیم‌سازی را با ویژگی کار ترکیب می‌کند.

ما در قسمت preprocessing، ابتدا حرف های عربی و چسبیده را به حروف فارسی تغییر دادیم و همچنین کاراکتر های اضافی مانند اعراب و نیم فاصله و اعداد انگلیسی و نشانه گذاری ها و ... را حذف کردیم تا دقت و همگرایی مدل بهتر شود که بصورت زیر کاراکتر های جایگزین شده و حذف شده قابل مشاهده است:

```
char_mappings = {'آ':'ا', 'آ':'ا', "ـا":"ا", 'ا':"ا", 'أ':'ا', 'ب':'ب', 'بـ':"ب", "بـ":"ب", "پ":"پ", 'ت':'ت', 'ت':'ت', 'تـ':'ت', "ة":"ت", 'ث':'ث',
                'ث':'ث', 'ج':'ج', 'ج':'ج', 'چ':'چ', 'چ':'چ', 'ح':'ح', 'ح':'ح', "خ":"خ", 'خ':'خ', 'غ':'غ', 'غ':'غ',
                'د':'د', 'د':"د", "ذ":'ذ', 'ذ':'ذ', 'ر':"ر", "ر":'ر', 'ز':'ز', 'ز':'ز',
                'س':'س', 'س':'س', 'س':"س", "س":'س', 'ش':'ش', 'ش':'ش', 'ص':'ص', 'ص':'ص', 'ض':'ض', 'ض':'ض',
                "ط":"ط", "ظ":"ظ", "ع":"ع", 'ع':'ع', 'غ':'غ', 'ف':'ف', 'ف':'ف', 'ف':'ف', 'ق':'ق', 'ق':'ک', 'ک':'ک', "ک":"ک", 'ك':'ک', "گ":"گ",
                "ل":"ل", 'لـ':'ل', 'ل':'ل', 'ل':'ل', "م":"م", 'م':'م', "مـ":'م', 'مـ':'م', 'ن':'ن', 'نـ':'ن', 'ن':'ن', "نـ":'ن',
                'و':'و', 'و':'و', "ۀ":'ه', 'ه':'ه', 'ۀ':'ه', 'ه':'ه', 'ة':'ه', 'ه':'ه', 'ھ':'ه',
                "ی":'ی', 'ی':'ی', 'ی':'ی', 'ی':'ی', "ی":'ی', 'یـ':"ی", "ی":'ی', 'ی':'ی',
                # '۱':'1','۲':'2', '۳':'3', '۴':'4', '۵':'5', '۶':'6', '۷':'7', '۸':'8', '۸':'8', '۹':'9'
                }
```

```
char_removals = [',', '(', ')', '!', '-', '،', '«', '…', '؛', '\n', '_', '،', ' ', '؟', '/', '-', '»',
                 '.', '"', '*', '،', '""', '””', '؟', '*', '!', '"', '؛', '"', '-', ':', '٪', '٪',
                 '\t', '\u202c', '\u202b', '\u200c','\xad', '\xa0'] + list(string.ascii_letters + string.digits)
```

۳.۴. تحلیل و نتیجه گیری

مدل Wav2Vec2ForCTC، با پایه و اساس آن در معماری Wav2Vec2 و نقطه موجود از پیش آموزش دیده، یک راه حل پیچیده برای ASR را نشان می دهد. ترکیبی از استخراج ویژگی، توجه به خود، از دست دادن CTC و مکانیسم‌های dropout به توانایی آن در رونویسی مؤثر زبان گفتاری به متن نوشتاری کمک می‌کند. پارامترهای قابل تنظیم امکان انطباق با وظایف مختلف را فراهم می کند و استراتژی آموزشی تعادل بین تعمیم و یادگیری ویژه کار را تضمین می کند. ارزیابی و اصلاح مداوم، نقاط قوت و زمینه‌های بالقوه مدل را برای بهبود در کاربردهای ASR در دنیای واقعی بیشتر روشن می‌کند.

- **میزان خطای مدل ASR و بررسی همبستگی این خطا به ویژگی ها**

مدل بیان شده توسط هایپرپارامتر ها و تعداد داده های مشخص مانند زیر آموزش دیده شده است:

```
***** Running training *****
  Num examples = 4,612
  Num Epochs = 8
  Instantaneous batch size per device = 8
  Total train batch size (w. parallel, distributed & accumulation) = 80
  Gradient Accumulation steps = 10
  Total optimization steps = 456
  Number of trainable parameters = 311,277,744
```

مقادیر هایپرپارامتر ها:

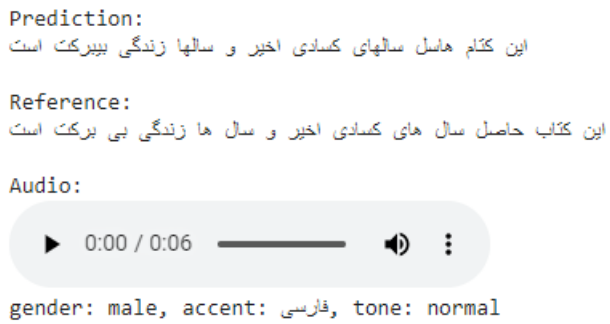
```
per_device_train_batch_size=8,
gradient_accumulation_steps=10,
evaluation_strategy="steps",
num_train_epochs=8,
fp16=True,
save_steps=100,
eval_steps=50,
logging_steps=10,
learning_rate=3e-4,
warmup_steps=10,
save_total_limit=1,
```

نتیجه نهایی آموزش مدل و مقادیر خطا در هر ۵۰ گام بصورت زیر میباشد:

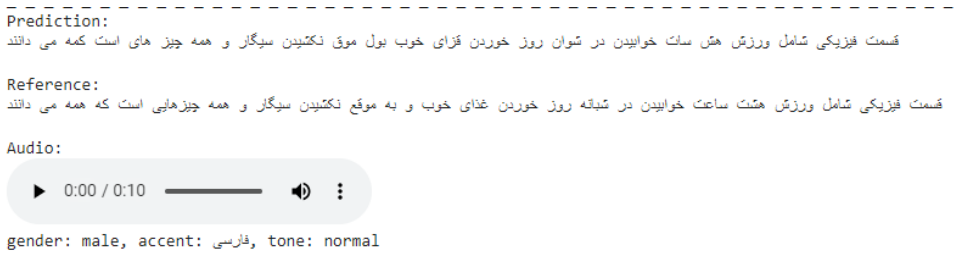
Step	Training Loss	Validation Loss	Wer
50	2.939000	2.927121	1.000000
100	2.901700	2.900964	1.000000
150	2.878100	2.876701	1.000000
200	2.861200	2.849467	1.000000
250	2.787500	2.695487	1.000575
300	2.000100	1.570530	0.971832
350	1.195600	0.860574	0.710846
400	0.948000	0.704943	0.612800
450	0.894100	0.661704	0.584824

در زیر چند نمونه از پیش‌بینی های صورت گرفته توسط مدل را مشاهده میکنیم:

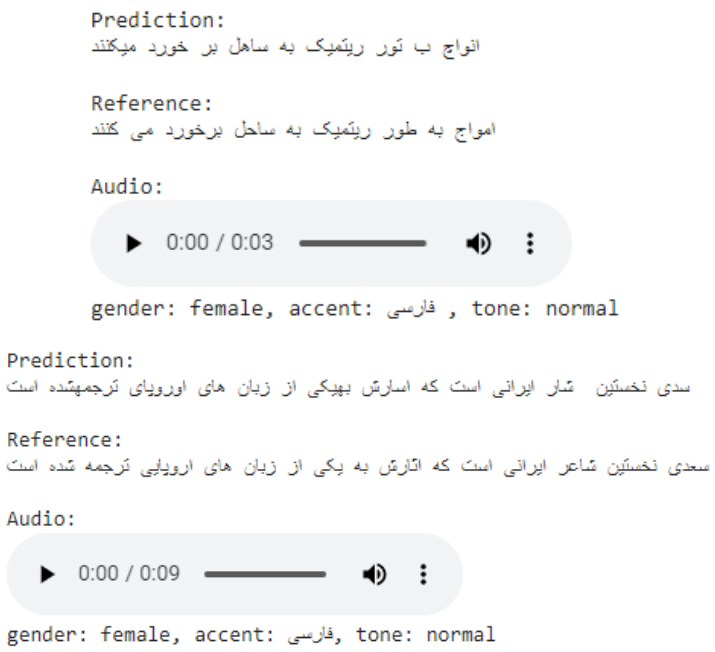
در برخی کلمات وا همان آ خوانده میشود و مدل را به اشتباه می اندازد همانطور که در مثال بالا، خاص بصورت خواس نوشته شده است.



همانطور که در این پیشبینی مشاهده میشود، بطور مثال، کلمه حاصل بصورت هاسل نوشته شده است که این میتوان به علت یکی بودن آوا و متفاوت بودن نوشتار باشد.



در این مثال به دلیل شباهت آوای ق و غ مدل به اشتباه افتاده است.



در این مثال نیز سدی و سعدی و همچنین شار و شاعر اشتباه شده که به دلیل کم و مشخص نبودن آوای "ع" در وویس ها میباشد. همچنین فاصله نیز رعایت نشده است.

با توجه به مثال های بیان شده و نتایج کلی میتوان گفت که مدل آموزش دیده شده، در چند مورد به اشتباه میافتد:

۱- زمانی که آوای حروف یکی بوده و املای آن ها متفاوت است. (مانند س ص ث)
۲- زمانی که استرس کلمه بصورت های متفاوت بیان میشود.
۳- زمانی که نوشتن و آوای کلمه متفاوت باشد (مانند وا و آ)
۴- زمانی که آوای حروف شبیه به هم باشد. (مانند ق غ)

بطور کلی میتوان نتیجه گرفت که در صورتی که ادای کلمات به درستی بیان نشده باشد، مدل به اشتباه میافتد که این موضوع با آموزش مدل با تعداد epoch های زیاد ممکن است حل شود.

مراجع:

[1] Raahul, A., Sapthagiri, R., Pankaj, K., & Vijayarajan, V. (2017). Voice based gender classification using machine learning. In IOP Conference Series: Materials Science and Engineering (Vol. 263, p. 042083). IOP Publishing. https://doi.org/10.1088/1757-899x/263/4/042083