دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

گزارش فاز دوم

پروژه نهایی درس یادگیری ماشین

تشخیص گویش و جنسیت از داده‌های صوتی

گروه ۴

ایلیا افضلی شماره دانشجویی: 810803069

نیما حسن خلق شماره دانشجویی: 810604076

ایلیا جعفری زاده شماره دانشجویی: 810803074

اساتید

محمدرضا ابوالقاسمی دهاقانی، مصطفی توسلی پور

بهار ۱۴۰۵

فهرست مطالب

۱ توضیح داده‌های ورودی و تعداد فایل‌های استفاده‌شده ۲

۲ توضیح روش‌های پاکسازی و پیش‌پردازش ۳
۱.۲ فیلتر کردن بر اساس مدت‌زمان ۳
۲.۲ پیش‌پردازش اولیه (استانداردسازی سیگنال) ۳
۳.۲ ارزیابی کیفیت سیگنال (SNR و VAD) ۳
۴.۲ فیلتر (Bandpass) برای داده‌های قابل اصلاح ۴

۳ دلیل انتخاب هر روش پیش‌پردازش ۵

۴ معرفی ویژگی‌های استخراج‌شده ۶
۱.۴ ضرایب کپسترال فرکانس ملی (MFCC) ۶
۲.۴ اسپکتروگرام لگاریتمی ملی (Log-Mel Spectrogram) ۶
۳.۴ فرکانس پایه / پیچ (Pitch / F0) ۶
۴.۴ ویژگی‌های کروما (Chroma Features) ۷
۵.۴ تضاد طیفی (Spectral Contrast) ۷
۶.۴ نرخ عبور از صفر (Zero Crossing Rate) ۷
۷.۴ انرژی میانگین مربعی (RMS Energy) ۸

۵ توضیح نحوه‌ی تبدیل هر فایل صوتی به بردار ویژگی ۹
۱.۵ روش تجمیع آماری ۹
۲.۵ ساختار بردار نهایی ۹

۶ تحلیل اولیه‌ی ویژگی‌ها و کیفیت دیتاست نهایی ۱۰
۱.۶ ساختار دیتاست نهایی ۱۰
۲.۶ توزیع کلاس‌ها ۱۰
۳.۶ تحلیل توزیع ویژگی‌های کلیدی ۱۰
۴.۶ تصویرسازی دوبعدی با PCA ۱۰

۷ ذکر چالش‌ها و تصمیم‌هایی که در طول این فاز گرفته شده است ۱۲
۱.۷ چالش دسترسی به فایل‌های پردازش‌شده ۱۲
۲.۷ چالش انتخاب تعداد ویژگی‌ها ۱۲
۳.۷ چالش کندی PYIN در استخراج Pitch ۱۲
۴.۷ چالش Session قطع‌شدن در گوگل کولب ۱۲
۵.۷ چالش عدم تعادل گویش مشهدی ۱۲

۱ توضیح داده‌های ورودی و تعداد فایل‌های استفاده‌شده

در این فاز، دیتاست خام نهایی که توسط تیم آموزشی منتشر شده بود، به عنوان داده‌های ورودی مورد استفاده قرار گرفت. برای جلوگیری از دانلود زمان‌بر داده‌ها، مصرف بی‌مورد حجم اینترنت و البته برای بهره‌گیری از قدرت پردازشی ابری، پوشه‌ی داده‌ها را به صورت میان‌بر (Shortcut) به گوگل درایو خود اضافه کردیم. سپس با ماونت کردن (Mount) درایو در محیط گوگل کولب (Google Colab)، فایل‌ها را مستقیما و بدون نیاز به انتقال محلی (Local) خواندیم.

ابتدا فایل final_metadata.csv را بارگذاری کردیم تا دید کلی نسبت به داده‌ها پیدا کنیم. در مجموع، دیتاست شامل ۴۳۱ فایل صوتی است. از آن‌جا که در فاز اول برخی گروه‌ها فرمت زمان‌بندی را به درستی رعایت نکرده بودند، تصمیم گرفتیم به ستون مدت‌زمان در متادیتا اعتماد نکنیم و مدت‌زمان واقعی (True Duration) هر فایل را مستقیما از خود فایل‌های صوتی استخراج کنیم.

آمار توصیفی مربوط به مدت‌زمان واقعی ۴۳۱ فایل صوتی در جدول ۱ خلاصه شده است. همان‌طور که مشخص است، میانگین طول فایل‌ها دقیقاً در حوالی ۶۰ ثانیه (یک دقیقه) قرار دارد.

جدول ۱: آمار توصیفی مدت‌زمان واقعی فایل‌های صوتی

شاخص آماری	مقدار (ثانیه)
تعداد کل	431
میانگین	60.08
انحراف معیار	4.53
کمینه	47.99
چارک اول (۲۵%)	57.94
میانه (۵۰%)	60.00
چارک سوم (۷۵%)	61.45
بیشینه	80.29

۲ توضیح روش‌های پاکسازی و پیش‌پردازش

در این بخش، (Pipeline) طراحی شده برای پیش‌پردازش فایل‌های صوتی را در سه فاز مجزا شرح می‌دهیم. هدف از این معماری، فیلتر کردن داده‌های پرت، استانداردسازی سیگنال‌ها و ارزیابی کمی کیفیت هر فایل پیش از استخراج ویژگی بود.

۱.۲ فیلتر کردن بر اساس مدت‌زمان

در قدم اول، برای رعایت استانداردهای جمع‌آوری داده، تنها فایل‌هایی را در دیتاست نگه داشتیم که مدت‌زمان واقعی آن‌ها به طور دقیق در بازه $[50, 70]$ ثانیه قرار داشت. با اعمال این فیلتر اولیه، فایل‌های بسیار کوتاه، بسیار بلند و همچنین فایل‌های خراب پیش از ورود به فازهای پردازشی سنگین حذف شدند.

۲.۲ پیش‌پردازش اولیه (استانداردسازی سیگنال)

تمام فایل‌های باقی‌مانده وارد یک مرحله پیش‌پردازش پایه شدند که شامل سه عملیات اصلی بود:

- **یکسان‌سازی نرخ نمونه‌برداری (Resampling):** نرخ نمونه‌برداری تمامی فایل‌ها به 16 kHz تغییر یافت. از نظر تئوری (قضیه نایکوئیست)، فرکانس 16 kHz به ما اجازه می‌دهد فرکانس‌های تا 8 kHz را به درستی بازسازی کنیم که این بازه تقریباً تمام اطلاعات آوایی مربوط به گفتار انسان را پوشش می‌دهد. این کار ضمن حفظ کامل اطلاعات مفید، بار محاسباتی استخراج ویژگی را به شدت کاهش داد.
- **حذف سکوت ابتدا و انتها:** با استفاده از تابع trim در کتابخانه librosa، بخش‌های بی‌صدای ابتدا و انتهای هر فایل (با آستانه ۲۰ دسی‌بل) حذف شد. این کار مانع از آن می‌شود که مدل در فازهای بعدی، نویز پس‌زمینه موجود در سکوت‌ها را به عنوان ویژگی‌های گویش یا جنسیت یاد بگیرد.
- **نرمال‌سازی صدا (Peak Normalization):** برای جبران تفاوت فاصله گویندگان تا میکروفون، دامنه سیگنال هر فایل بر مقدار بیشینه مطلق آن تقسیم شد تا تمامی مقادیر در بازه $[-1, 1]$ قرار گیرند. این کار باعث می‌شود شدت صدای تمامی فایل‌ها در یک مقیاس برابر و قابل مقایسه باشد.

۳.۲ ارزیابی کیفیت سیگنال (SNR و VAD)

پس از پیش‌پردازش اولیه، کیفیت هر فایل را با دو معیار کمی محاسبه کردیم تا فایل‌های با کیفیت پایین را شناسایی کنیم. این مرحله به ما کمک کرد فایل‌ها را به سه دسته OK، Fixable و Trash تقسیم کنیم:

- **تشخیص فعالیت صوتی (Voice Activity Detection - VAD):** برای اطمینان از اینکه فایل صوتی واقعاً حاوی گفتار مفید است، از مدل شبکه‌های عصبی از پیش آموزش‌دیده Silero VAD استفاده کردیم. این مدل، Timestamps بخش‌های حاوی صدای انسان را استخراج می‌کند.

درصد گفتار موجود در فایل به صورت زیر محاسبه شد:

$$\% \text{ VAD} = \frac{\sum(t_{\text{end}} - t_{\text{start}})}{\text{Total Length}} \times 100$$

- **نسبت سیگنال به نویز (SNR):** برای محاسبه میزان نویز محیط، از انرژی RMS (Root Mean Square) در فریم‌های مختلف سیگنال استفاده کردیم. فریم‌ها بر اساس میزان انرژی مرتب شدند؛ میانگین ۱۰ درصد بالاترین انرژی‌ها به عنوان قدرت سیگنال گوینده (Speech Signal) و میانگین ۱۰ درصد پایین‌ترین انرژی‌ها به عنوان قدرت نویز پس‌زمینه (Background Noise) در نظر گرفته شد. فرمول SNR بر حسب دسی‌بل به شکل زیر پیاده‌سازی شد:

$$\text{SNR}_{\text{dB}} = 20 \log_{10} \left(\frac{\mu_{\text{RMS ۱۰\% top}}}{\mu_{\text{RMS ۱۰\% bottom}}} \right)$$

۴.۲ فیلتر (Bandpass) برای داده‌های قابل اصلاح

فایل‌هایی که در مرحله‌ی ارزیابی کیفیت در دسته‌ی Fixable قرار گرفتند، دارای نویز پس‌زمینه یا مشکلات فرکانسی بودند اما همچنان اطلاعات زبانی مفیدی برای یادگیری ماشین داشتند. برای پاکسازی این سیگنال‌ها و حذف فرکانس‌های مزاحم (مانند نویز های محیطی)، از یک (Butterworth Bandpass Filter) با درجه ۵ استفاده کردیم که به کمک کتابخانه‌ی scipy روی این دسته از فایل‌ها اعمال شد.

فرکانس (Lowcut) روی 80 Hz تنظیم شد تا نویزهای محیطی سنگین (مانند صدای ترافیک، باد یا لرزش میکروفون) حذف شوند، بدون اینکه به فرکانس‌های پایه‌ی صدای انسان آسیبی برسد. از سوی دیگر، فرکانس (Highcut) روی 7500 Hz قرار گرفت. این انتخاب علاوه بر حذف نویزهای فرکانس بالا (Hiss)، تضمین می‌کند که سیگنال ما اکیداً پایین‌تر از مرز فرکانس نایکوئیست که برای فایل‌های ما 8000 Hz است باقی بماند. در نهایت، با عبور داده‌ها از این پایپلاین، موفق شدیم ۳۷۹ فایل باکیفیت و استاندارد را برای فاز استخراج ویژگی ذخیره کنیم.

نکته در خصوص نویزهای گذرا:
در طول بررسی شنیداری داده‌ها متوجه شدیم که برخی فایل‌ها دارای نویزهای کوتاه و گذرا (Transient Noises) در پس‌زمینه هستند (مانند صدای تق‌تق یا خرد کردن روی تخته‌ی آشپزخانه). تصمیم گرفتیم این نویزهای لحظه‌ای را به شکل تهاجمی حذف نکنیم؛ چرا که تلاش برای بریدن آن‌ها باعث ایجاد سکوت یا قطعی در میانه‌ی گفتار می‌شود و پیوستگی سیگنال را از بین می‌برد. از طرفی، استفاده از الگوریتم‌های پیچیده‌ترِ حذف نویز باعث اصطلاحاً «رباتی شدن» صدا (Robotic Artifacts) می‌شود که ویژگی‌های طبیعی آوایی را تخریب می‌کند. از آن‌جا که مدل‌های یادگیری ماشین معمولاً نسبت به نویزهای لحظه‌ای و غیرالگو مقاومت خوبی دارند، حفظ حالت طبیعی و پیوسته‌ی گفتار را برای تشخیص گویش‌ها در اولویت قرار دادیم.

۳ دلیل انتخاب هر روش پیش‌پردازش

در این مرحله، هدف ما این بود که مدل فقط روی ویژگی‌های اصلی گویش و جنسیت تمرکز کند و درگیر اطلاعات بی‌ارزش یا خطاهای ضبط نشود. تصمیمات ما برای انتخاب هر روش بر اساس مشاهدات زیر بود:

- **فیلتر بازه زمانی (۵۰ تا ۷۰ ثانیه):** طبق دستورالعمل پروژه، فایل‌ها نباید کمتر از ۵۰ یا بیشتر از ۷۰ ثانیه می‌بودند. اعمال این فیلتر باعث شد تا داده‌های پرت (خیلی کوتاه یا خیلی طولانی) که می‌توانستند در فرآیند استخراج ویژگی وزن نامتعارفی بگیرند، حذف شوند.
- **حذف سکوت و Normalize کردن صدا:** سکوت‌های ابتدا و انتهای فایل هیچ دیتای زبانی مفیدی ندارند و فقط مدل را گیج می‌کنند. از طرفی، چون فایل‌ها از منابع مختلفی جمع‌آوری شده بودند، ولوم صدای گویندگان به شدت متفاوت بود (یکی بسیار بلند و دیگری بسیار ضعیف). با Normalize کردن فایل‌ها، مطمئن شدیم که بلندی صدا به عنوان یک ویژگی غلط توسط مدل یاد گرفته نمی‌شود.
- **استفاده از Sampling Rate روی 16k:** این کار دو دلیل اصلی داشت. اول اینکه حجم فایل‌ها و بار محاسباتی سیستم به شدت پایین می‌آید. دوم اینکه محدوده فرکانسی صدای انسان به طور کامل در همین نرخ پوشش داده می‌شود و ما اصلاً نیازی نداریم مدل را روی فرکانس‌های بالایی که هیچ ربطی به صدای انسان ندارند Train کنیم.
- **ترکیب SNR و VAD:** وقتی به صورت رندوم تعدادی از ویس‌های متادیتا را بررسی کردم، متوجه شدم بعضی از آن‌ها نویز زیادی دارند یا پر از سکوت‌های طولانی بین کلمات هستند و عملاً برای آموزش مدل به درد نمی‌خورند. به همین دلیل تصمیم گرفتیم از SNR (برای سنجش میزان نویز) و VAD (برای تشخیص درصد واقعی صحبت کردن انسان در طول فایل) در کنار هم استفاده کنیم. این ترکیب به ما کمک کرد تا به صورت خودکار و دقیق بفهمیم کدام ویس‌ها ارزش نگه داشتن دارند و کدام‌ها را باید حذف کنیم.
- **استفاده از Bandpass:** برای ویس‌هایی که نویز داشتند اما قابل اصلاح تشخیص داده شدند، از فیلتر Bandpass استفاده کردیم. بزرگترین مزیت این فیلتر این است که نویزهای اضافی و مزاحم را حذف می‌کند اما به جریان طبیعی و آوایی صدا آسیبی نمی‌زند؛ حفظ این پیوستگی و حالت طبیعی صدا برای تشخیص درست گویش‌ها توسط مدل ضروری است.

۴ معرفی ویژگی‌های استخراج‌شده

پس از پاکسازی داده‌ها، هدف ما تبدیل هر فایل صوتی به یک بردار عددی با طول ثابت بود. برای این منظور، هفت گروه ویژگی مکمل از هر فایل استخراج شد. انتخاب این ویژگی‌ها بر اساس ماهیت پنج گویش مورد بررسی (ترکی، کردی، مشهدی، اصفهانی، گیلکی) صورت گرفت؛ چرا که تفاوت‌های اصلی این گویش‌ها در کیفیت واکه‌ها، آهنگ گفتار و ساختار آوایی نهفته است. در مجموع ۹۹ ویژگی عددی از هر فایل استخراج شد که در ادامه هر گروه معرفی می‌شود.

۱.۴ ضرایب کپسترال فرکانس ملی (MFCC)

نحوه استخراج: سیگنال صوتی به فریم‌های کوتاه تقسیم می‌شود، روی هر فریم DFT اعمال می‌گردد، طیف توان روی مقیاس ملی نگاشته شده، لگاریتم گرفته می‌شود و در نهایت با DCT فشرده‌سازی می‌شود. ۱۳ ضریب از هر فریم محاسبه شد.

چرا مفید است: MFCC فشرده‌ترین نمایش شکل دستگاه صوتی (حنجره و حلق) است. گویش‌های مختلف در کیفیت واکه‌ها و تلفظ همخوان‌ها با یکدیگر تفاوت دارند و این تفاوت‌ها مستقیماً در ضرایب پایین MFCC کدگذاری می‌شوند. همچنین تفاوت طول دستگاه صوتی بین زن و مرد نیز در پوش کلی MFCC بازتاب می‌یابد.

مزایا و محدودیت‌ها: فشرده، محکم و کم‌هزینه از نظر محاسباتی است. اما اطلاعات فاز سیگنال و دینامیک‌های ریز زمانی را نادیده می‌گیرد.

تعداد ویژگی: ۱۳ ضریب × (میانگین + انحراف معیار) = ۲۶ ویژگی

۲.۴ اسپکتروگرام لگاریتمی ملی (Log-Mel Spectrogram)

نحوه استخراج: پس از محاسبه STFT، طیف توان روی ۱۳ فیلتربانک مثلثی در مقیاس ملی نگاشته شده و به دسی‌بل تبدیل می‌شود.

چرا مفید است: برخلاف MFCC، این ویژگی فشرده‌سازی DCT را ندارد و جزئیات بیشتری از توزیع انرژی طیفی را حفظ می‌کند. گویش‌های مختلف شکل‌های طیفی مشخصه‌ای در واکه‌هایشان دارند که Log-Mel آن‌ها را نگه می‌دارد. تفاوت فرکانس پایه بین زن و مرد نیز در باندهای فرکانسی پایین این ویژگی نمایان است.

چرا ۱۳ باند به جای ۴۰ یا ۱۲۸: دیتاست این پروژه حداکثر حدود ۳۰۰ فایل دارد. استفاده از ۴۰ باند به معنای ۸۰ ویژگی از یک استخراج‌گر بود که نسبت ویژگی به نمونه را به شدت بالا می‌برد و خطر Overfitting در فاز سوم را افزایش می‌داد. ۱۳ باند تعادل مناسبی بین جزئیات طیفی و پیچیدگی مدل فراهم می‌کند.

تعداد ویژگی: ۱۳ باند × (میانگین + انحراف معیار) = ۲۶ ویژگی

۳.۴ فرکانس پایه / پیچ (Pitch / F0)

نحوه استخراج: با استفاده از الگوریتم PYIN (نسخه احتمالاتی YIN)، فرکانس پایه فریم‌به‌فریم تخمین زده می‌شود. این الگوریتم همچنین یک پرچم صدادار/بی‌صدا برای هر فریم برمی‌گرداند. آمارها فقط روی فریم‌های صدادار محاسبه شدند تا میانگین به وسیله سکوت‌ها آلوده نشود.

چرا مفید است: F0 مستقیم‌ترین نشانگر جنسیت است. گویندگان زن در فارسی معمولاً در بازه ۱۸۰ تا ۲۵۵ هرتز و مردان در بازه ۸۵ تا ۱۶۵ هرتز قرار دارند. علاوه بر این، گویش‌های ایرانی الگوهای اینتوناسیون مشخصی دارند؛ برای مثال گیلکی به خاطر آهنگ صعودی مشخص و اصفهانی به خاطر الگوی موسیقایی خود شناخته شده‌اند.

مزایا و محدودیت‌ها: بهترین ویژگی برای تشخیص جنسیت و قابل تفسیر است. اما PYIN کندترین الگوریتم در این Pipeline است و ممکن است روی فایل‌های پر نویز نتایج ضعیفی بدهد.

تعداد ویژگی: ۴ ویژگی (میانگین، انحراف معیار، کمینه، بیشینه روی فریم‌های صدادار)

۴.۴ ویژگی‌های کروما (Chroma Features)

نحوه استخراج: طیف توان روی ۱۲ کلاس گام موسیقی (دو، دودیز، ر، ...) نگاشته می‌شود. هر Bin مجموع انرژی مرتبط با آن کلاس گام را بدون توجه به اکتاو نگه می‌دارد.

چرا مفید است: این ویژگی به طور مستقیم برای تمایز گویش‌های این پروژه مفید است. گویش‌های ایرانی در الگوهای اینتوناسیون خود تفاوت‌های چشمگیری دارند؛ گیلکی دارای الگوهای صعودی منحصربه‌فرد، اصفهانی دارای الگوی موسیقایی و مشهدی دارای الگوی مسطح‌باافت است. این عادات ملودیک منجر به توزیع‌های پایدار انرژی در کلاس‌های گامی می‌شود که کروما آن را ثبت می‌کند. علاوه بر این، کروما نسبت به تغییرات تمبر مقاوم است که آن را مکمل خوبی برای MFCC می‌کند.

مزایا و محدودیت‌ها: حساس به گویش و مقاوم در برابر تفاوت‌های گوینده است. اما اصالتاً برای موسیقی طراحی شده و ممکن است هارمونیک‌های نویز پس‌زمینه را با الگوی گفتار اشتباه بگیرد.

تعداد ویژگی: ۱۲ کلاس گام × (میانگین + انحراف معیار) = ۲۴ ویژگی

۵.۴ تضاد طیفی (Spectral Contrast)

نحوه استخراج: طیف به ۶ زیرباند تقسیم می‌شود و برای هر باند، تفاوت بین انرژی پیک‌ها (قله‌های طیفی) و دره‌ها (کمینه‌های طیفی) محاسبه می‌شود.

چرا مفید است: گفتار هارمونیک‌محور (مثل گویش‌های با واکه‌های طولانی) تضاد بالایی بین قله‌ها و دره‌ها دارد، در حالی که گفتار با سایش بیشتر تضاد کمتری نشان می‌دهد. موجودی آوایی گویش‌های مختلف الگوهای ثابتی در این زیرباندها ایجاد می‌کند.

مزایا و محدودیت‌ها: ساختار هارمونیک را فراتر از شکل طیفی ساده ثبت می‌کند. اما نسبت به نویز در دره‌های طیفی حساس است.

تعداد ویژگی: ۷ باند × (میانگین + انحراف معیار) = ۱۴ ویژگی

۶.۴ نرخ عبور از صفر (Zero Crossing Rate)

نحوه استخراج: تعداد دفعاتی که دامنه موج صوتی در هر فریم از صفر عبور می‌کند شمارش و میانگین‌گیری می‌شود.

چرا مفید است: نرخ عبور از صفر بالا نشان‌دهنده گفتار پر از سایش یا بی‌صدا است، در حالی که نرخ پایین نشانه بخش‌های صدادار و هارمونیک است. گویش‌ها در تراکم همخوان‌های سایشی و خوشه‌های همخوانی با یکدیگر تفاوت دارند.

مزایا و محدودیت‌ها: بسیار سریع است و دیدگاه مکملی بر اساس حالت صداداری فراهم می‌کند. اما تنها یک آمار کلی است و جزئیات زمانی را از دست می‌دهد.

تعداد ویژگی: ۲ ویژگی (میانگین و انحراف معیار)

۷.۴ انرژی میانگین مربعی (RMS Energy)

نحوه استخراج: دامنه RMS در هر فریم کوتاه محاسبه و سری زمانی انرژی حاصل خلاصه می‌شود.

چرا مفید است: سبک گفتاری و تأکید پروزودیک در گویش‌های مختلف متفاوت است. دامنه دینامیکی (انحراف معیار) و اوج بلندی (بیشینه) نشان می‌دهند که گوینده چقدر شدت صدایش را در طول زمان تغییر می‌دهد که یک نشانه پروزودیک است.

مزایا و محدودیت‌ها: ساده، سریع و مکمل ویژگی‌های طیفی است. اما از آنجا که در مرحله پیش‌پردازش نرمال‌سازی صدا انجام شده، واریانس مطلق بین فایل‌ها کاهش یافته و اهمیت این ویژگی نسبت به حالت بدون نرمال‌سازی کمتر است.

تعداد ویژگی: ۳ ویژگی (میانگین، انحراف معیار، بیشینه)

جدول ۲: خلاصه ویژگی‌های استخراج‌شده

گروه ویژگی	هدف اصلی	تعداد ویژگی
MFCC	گویش + جنسیت	۲۶
Log-Mel Spectrogram	گویش + جنسیت	۲۶
Pitch / F0	جنسیت + پروزودی	۴
Chroma	گویش (اینتوناسیون)	۲۴
Spectral Contrast	گویش (هارمونیک)	۱۴
Zero Crossing Rate	گویش (آواشناسی)	۲
RMS Energy	پروزودی	۳
جمع کل		**۹۹**

۵ توضیح نحوه‌ی تبدیل هر فایل صوتی به بردار ویژگی

مدل‌های یادگیری ماشین به ورودی با ابعاد ثابت نیاز دارند، اما اکثر ویژگی‌های صوتی مانند MFCC و Log-Mel Spectrogram به صورت ماتریس دوبعدی (ضرایب × فریم زمانی) استخراج می‌شوند. طول این ماتریس به مدت‌زمان فایل بستگی دارد و بین فایل‌ها متفاوت است. برای حل این مشکل، از روش **تجمیع آماری** (Statistical Aggregation) استفاده کردیم.

۱.۵ روش تجمیع آماری

برای هر ویژگی که در طول زمان تغییر می‌کند، دو آماره‌ی **میانگین** و **انحراف معیار** روی بُعد زمان محاسبه می‌شود. این دو آماره اطلاعات مکملی را ثبت می‌کنند:

- **میانگین:** ویژگی‌های پایدار و بلندمدت گوینده را می‌گیرد؛ مثلاً میانگین MFCC-2 نمایانگر شکل کلی دستگاه صوتی است.
- **انحراف معیار:** میزان تغییرپذیری و پویایی در طول گفتار را ثبت می‌کند؛ مثلاً انحراف معیار Pitch نشان‌دهنده تنوع اینتوناسیون است.

به عنوان مثال، برای MFCC با ۱۳ ضریب:

$$\text{بردار MFCC} = [\mu_1, \sigma_1, \mu_2, \sigma_2, \ldots, \mu_{13}, \sigma_{13}] \in \mathbb{R}^{26}$$

۲.۵ ساختار بردار نهایی

با اعمال این روش روی تمام گروه‌های ویژگی، هر فایل صوتی به یک بردار ۹۹ بعدی تبدیل می‌شود:

$$\mathbf{x} = [\underbrace{\text{MFCC}}_{26}, \underbrace{\text{Log-Mel}}_{26}, \underbrace{\text{Pitch}}_{4}, \underbrace{\text{Chroma}}_{24}, \underbrace{\text{SC}}_{14}, \underbrace{\text{ZCR}}_{2}, \underbrace{\text{RMS}}_{3}] \in \mathbb{R}^{99}$$

این بردار به همراه چهار ستون متادیتا (نام فایل، گویش، جنسیت، شناسه گوینده) یک سطر در دیتاست نهایی را تشکیل می‌دهد. در فاز سوم، ستون‌های گویش و جنسیت به عنوان برچسب هدف (Target Label) و ۹۹ ویژگی عددی به عنوان ماتریس ورودی (Feature Matrix) مدل‌های طبقه‌بندی و خوشه‌بندی مورد استفاده قرار خواهند گرفت.

۶ تحلیل اولیه‌ی ویژگی‌ها و کیفیت دیتاست نهایی

۱.۶ ساختار دیتاست نهایی

پس از استخراج موفقیت‌آمیز ویژگی‌ها، دیتاست نهایی شامل ۳۷۹ سطر (فایل صوتی) و ۱۰۳ ستون (۴ متادیتا + ۹۹ ویژگی) است. هیچ مقدار گمشده‌ای در ماتریس ویژگی‌ها وجود ندارد.

۲.۶ توزیع کلاس‌ها

توزیع داده‌ها بین گویش‌ها و جنسیت‌ها در جدول ۳ خلاصه شده است. همان‌طور که مشخص است، توازن نسبتاً خوبی بین فایل‌های مربوط به گویندگان زن و مرد وجود دارد که برای آموزش مدل‌های طبقه‌بندی بسیار مطلوب است.

جدول ۳: توزیع فایل‌های صوتی بر اساس گویش و جنسیت

گویش	تعداد زن	تعداد مرد	جمع
اصفهانی (Isfahani)	۳۱	۳۱	۶۲
ترکی (Torki)	۵۶	۵۶	۱۱۲
کردی (Kordi)	۴۵	۴۶	۹۱
مشهدی (Mashhadi)	۱۶	۱۱	۲۷
گیلکی (Gilaki)	۵۵	۳۲	۸۷
جمع کل	**۲۰۳**	**۱۷۶**	**۳۷۹**

۳.۶ تحلیل توزیع ویژگی‌های کلیدی

دو نمودار اصلی برای بررسی کیفیت ویژگی‌ها رسم شد:

- **توزیع Pitch بر اساس جنسیت:** نمودار هیستوگرام میانگین F0 نشان داد که میانگین فرکانس پایه گویندگان زن به وضوح بالاتر از مردان است. این جداپذیری واضح تأیید می‌کند که ویژگی Pitch اطلاعات ارزشمندی برای تشخیص جنسیت دارد.
- **توزیع MFCC-1 بر اساس گویش:** نمودار Boxplot نشان داد که گویش‌های مختلف میانه‌ها و دامنه‌های متفاوتی برای این ضریب دارند که نشان‌دهنده تفاوت در انرژی کلی و ساختار طیفی گویش‌هاست.

۴.۶ تصویرسازی دوبعدی با PCA

برای بررسی اولیه جداپذیری داده‌ها، PCA با دو مؤلفه اصلی روی ماتریس ویژگی‌های استانداردشده اعمال شد. نتایج نشان داد که دو مؤلفه اول مجموعاً درصد قابل توجهی از واریانس کل داده‌ها را توضیح می‌دهند.

در نمودار رنگ‌بندی‌شده بر اساس جنسیت، خوشه‌بندی نسبتاً مشخصی بین زن و مرد قابل مشاهده بود که نشان می‌دهد ویژگی‌های استخراج‌شده اطلاعات کافی برای تمایز جنسیت دارند. در نمودار رنگ‌بندی‌شده بر اساس گویش، هرچند هنوز همپوشانی‌هایی بین گویش‌ها وجود داشت، اما برخی گویش‌ها مانند گیلکی و اصفهانی تمایل به تشکیل خوشه‌های جداگانه‌تری داشتند.

نکته در خصوص همپوشانی گویش‌ها در PCA:
همپوشانی مشاهده‌شده در نمودار PCA گویش‌ها لزوماً به معنای ضعف ویژگی‌ها نیست. PCA یک تصویر خطی است که تنها ۲ بُعد از ۹۹ بُعد را نشان می‌دهد. مدل‌های غیرخطی مانند SVM با کرنل RBF یا جنگل تصادفی در فاز سوم می‌توانند از کل ۹۹ بُعد برای جداسازی استفاده کنند و عملکرد بهتری ارائه دهند.

۷ ذکر چالش‌ها و تصمیم‌هایی که در طول این فاز گرفته شده است

۱.۷ چالش دسترسی به فایل‌های پردازش‌شده

از آنجا که این پروژه به صورت گروهی انجام می‌شود و هر عضو گروه بخشی از کار را بر عهده دارد، فایل‌های صوتی پردازش‌شده در فولدر مشترک گوگل درایو موجود نبودند. برای حل این مشکل، Pipeline استخراج ویژگی به گونه‌ای طراحی شد که ابتدا دنبال فایل پردازش‌شده می‌گردد و در صورت نبود آن، مستقیماً از فایل خام دیتاست اصلی استفاده می‌کند. این تصمیم کار را پیوسته نگه داشت و نیاز به هماهنگی همزمان بین اعضای گروه را برطرف کرد.

۲.۷ چالش انتخاب تعداد ویژگی‌ها

یکی از تصمیم‌های مهم این فاز، انتخاب تعداد باندهای Log-Mel Spectrogram بود. استفاده از ۴۰ یا ۱۲۸ باند (که در بسیاری از مقالات رایج است) برای دیتاست‌های بزرگ مناسب است، اما برای این پروژه با حدود ۳۰۰ تا ۴۰۰ فایل، می‌توانست منجر به Overfitting در فاز سوم شود. تصمیم گرفتیم تعداد باندها را به ۱۳ کاهش دهیم تا نسبت تعداد ویژگی به نمونه در محدوده مناسبی باقی بماند.

۳.۷ چالش کندی PYIN در استخراج Pitch

الگوریتم PYIN برای استخراج دقیق فرکانس پایه طراحی شده اما نسبت به سایر ویژگی‌ها به زمان بیشتری نیاز دارد. با توجه به اینکه Pitch مستقیم‌ترین ویژگی برای تشخیص جنسیت است، تصمیم گرفتیم آن را حفظ کنیم و کندی اجرا را بپذیریم. برای فایل‌هایی که هیچ فریم صوتی در آن‌ها تشخیص داده نشد، مقدار صفر برگردانده شد تا Pipeline بدون خطا ادامه یابد.

۴.۷ چالش Session قطع‌شدن در گوگل کولب

در طول اجرای Pipeline، به دلیل قطع ارتباط مرورگر با سرور Colab، Session ریست شد و متغیرها از حافظه پاک شدند. برای جلوگیری از تکرار این مشکل، تمام خروجی‌های میانی (از جمله processed_meta-data_handoff.csv و فایل ویژگی‌های نهایی) بلافاصله پس از محاسبه روی گوگل درایو ذخیره شدند تا در صورت قطع مجدد، نیازی به اجرای مجدد از ابتدا نباشد.

۵.۷ چالش عدم تعادل گویش مشهدی

همان‌طور که در جدول توزیع مشاهده می‌شود، گویش مشهدی تنها ۲۷ فایل دارد در حالی که ترکی ۱۱۲ فایل دارد. این عدم تعادل می‌تواند در فاز سوم باعث شود مدل به سمت گویش‌های با نمونه بیشتر تمایل پیدا کند. این چالش را در گزارش فاز سوم به عنوان یک متغیر مهم در ارزیابی مدل‌ها مد نظر قرار خواهیم داد و از تکنیک‌هایی مانند Weighted Loss یا Stratified Cross-Validation استفاده خواهیم کرد.

مراجع

[۱] Silero Team, ”Silero VAD: pre-trained enterprise-grade Voice Activity Detector”, GitHub, https://github.com/snakers4/silero-vad

[۲] McFee, B. et al., ”librosa: Audio and Music Signal Analysis in Python”, Proceedings of the 14th Python in Science Conference, 2015.

[۳] Davis, S. and Mermelstein, P., ”Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences”, IEEE Transactions on Acoustics, Speech, and Signal Processing, 1980.

[۴] Mauch, M. and Dixon, S., ”pYIN: A Fundamental Frequency Estimator Using Probabilistic Threshold Distributions”, ICASSP, 2014.