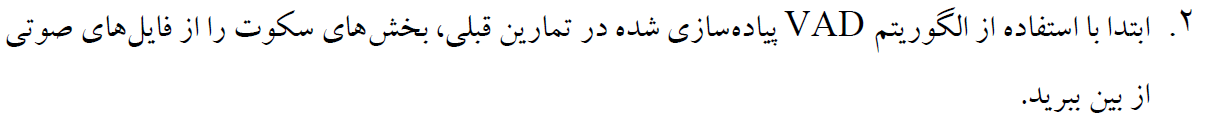
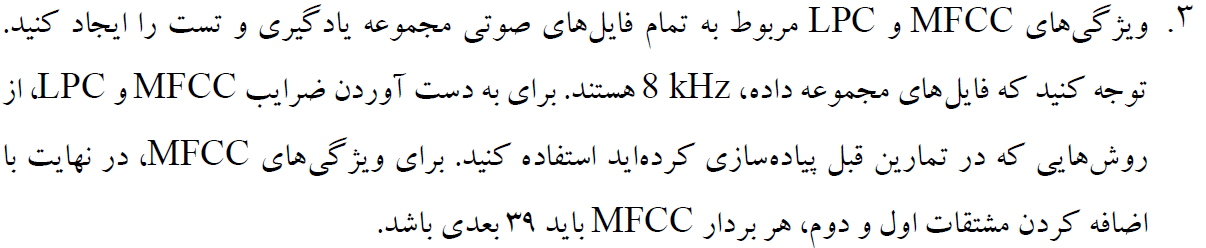


در بخش Preprocess functions توابع زیر تعریف شده است:

* Read-voice: تابعی برای خواندن یک فایل صوتی.
* Pre\_emphasis : عملیات پیش تاکید در پیش پردازش.
* Framming: عملیات فریم‌بندی برای یک فایل صوتی.
* Windowing: تابعی برای پنجره‌گذاری در یک فریم.
* Preprocess: تابع پیش‌پردازش بر روی داده‌ی صوتی استخراج شده. (پیش‌تاکید، فریم‌بندی و پنجره‌گذاری.)

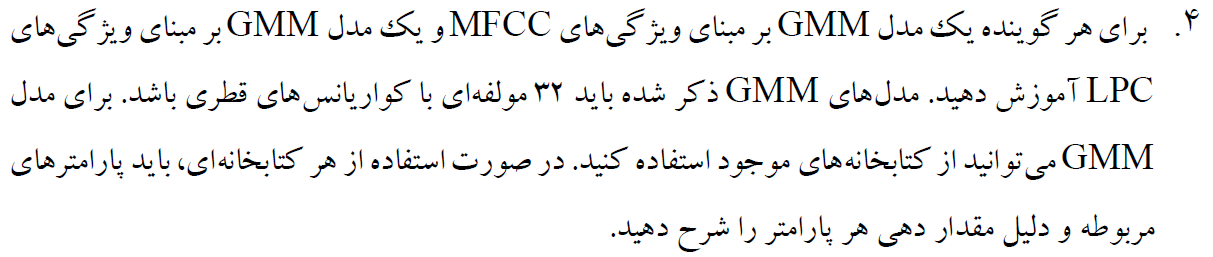


در بخش VAD function تابعی تحت عنوان VAD وجود دارد که تعدادی فریم در ورودی گرفته و بر اساس لگاریتم انرژی و تعیین یک آستانه (مقدار -6)، فریم‌های سکوت را حذف می‌کند.



در بخش LPC and MFCC features functions با استفاده از کتابخانه‌های spectrum و librosa، ویژگی‌های LPC و MFCC برای هر فریم بدست می‌آید. (توابع get\_lpc، get\_mfcc). همچنین دو تابع lpc\_feature\_of\_frames و mfcc\_feature\_of\_frames به صورت بهینه، ویژگی‌های مورد نظر را برای فریم‌های استخراج شده، بدست می‌آورد. تعداد بعد ویژگی‌های LPC 39 تا، ویژگی‌های MFCC (شامل خود ضرایب و مشتقات درجه یک و دو) نیز دارای 39 بعد می‌باشد.

در بخش read data and make dataframe نیز مسیر فایل‌های موجود شناسایی و خوانده شده است و برای هر صوت خوانده شده، پیش‌پردازش (pre-emphasis، framing، windowing و VAD) انجام گرفته، سپس برای هر فایل ویژگی‌های ذکر شده در بخش قبل، استخراج شده است. در انتها دو دیتافریم train\_df و test\_df بدست آورده شد. هر کدام این دیتافریم‌ها شامل speaker، data، آرایه‌ی lpc و آرایه‌ی mfcc است.



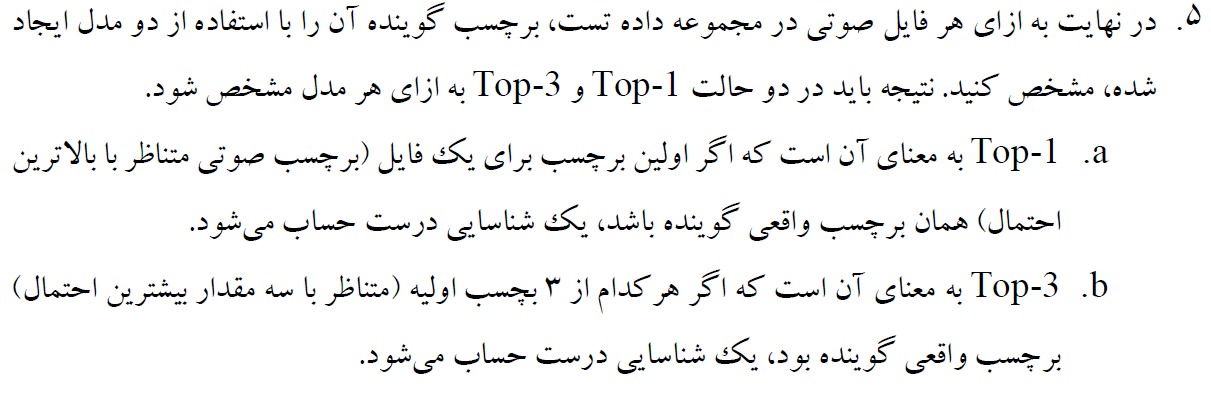
در بخش GMM، توابع زیر پیاده سازی شده است:

* get\_LPC\_GMM\_model\_of\_speaker: تابعی که دیتافریم آموزشی مربوط به یک گوینده را گرفته و بر اساس ویژگی‌های lpc آن مدل می‌سازد. برای 32 مولفه‌ای بودن باید پارامتر n\_component (تعداد گوسین( آن را برابر با 32 و برای قطری بودن از covariance\_type = 'diag' استفاده میکنیم. همچنین آموزش آن در 100 دور انجام میشود.
* get\_MFCC\_GMM\_model\_of\_speaker: تابعی که دیتافریم آموزشی مربوط به یک گوینده را گرفته و بر اساس ویژگی‌های MFCC آن مدل می‌سازد. برای 32 مولفه‌ای بودن باید پارامتر n\_component (تعداد گوسین( آن را برابر با 32 و برای قطری بودن از covariance\_type = 'diag' استفاده میکنیم. همچنین آموزش آن در 100 دور انجام میشود.

در دو تابع فوق فریم‌ها همگی با هم concatenate میشوند.

* Get\_GMMs: مدل‌های GMM برای هر یک از ویژگی‌های LPC و MFCC برای تمامی گویندگان (20 مدل برای هر ویژگی که می‌شود 40 مدل) بر می‌گرداند.

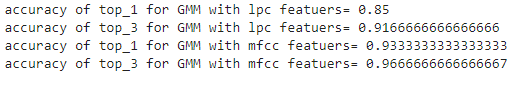
در این بخش مدل‌ها ساخته شده و در قالب لیست‌هایی نگه داشته شده است.



در بخش predict speakers، تابع Predict و Predict\_top\_3 پیاده‌سازی شده است که در آن با توجه به score داده شده برای هر مدل و برای هر گوینده‌ی آزمایشی، مقادیر sort شده و سه تای بهترین برگردانده شده است.سپس برای هر یک از سطر‌های test\_df پیش‌بینی انجام شده است، سپس دیتافریمی تحت عنوان pred\_df که شامل ستون‌های real و lpc\_pred و mfcc\_pred برای تمامی سطر‌های test\_df ساخته شده است.

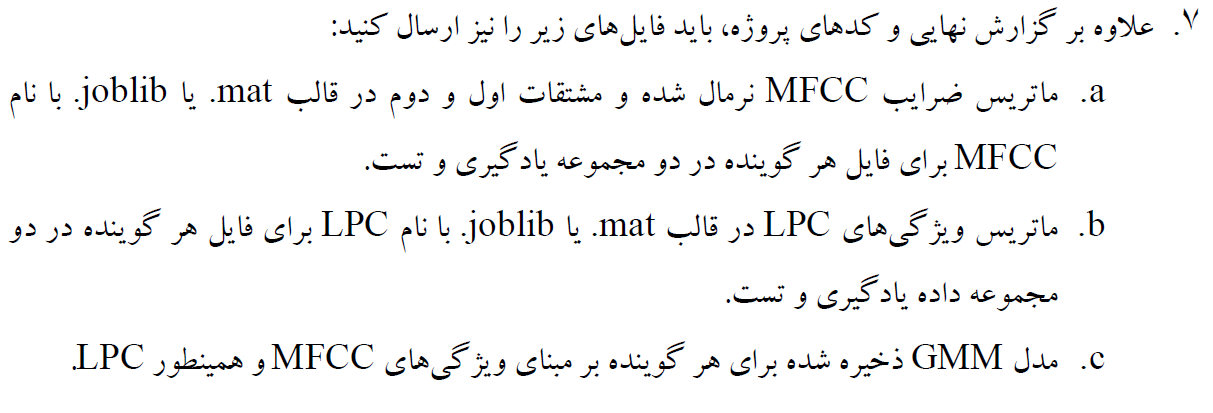


در بخش predict speakers، تابع accuracy نوشته شده است که در آن دقت GMM برای ویژگی و top\_k ورودی محاسبه می‌شود. نتایج بدست آمده برای هر کدام از حالات ذکر شده به صورت زیر است:



همانطور که نشان داده شده است ویژگی‌های MFCC از ویژگی‌های LPC برای تشخیص گوینده کارآمدتر بوده است.

همچنین اگر مدل نتوانسته در گزینه اول تشخیص دهد به احتمال خیلی زیاد در سه گزینه‌ی اول انتخابی‌اش توانسته درست پیش‌بینی کند.



در بخش save matrixes ماتریس ویژگی‌های MFCC، LPC برای train و test ذخیره شده است. این فایل‌ها با نام مناسب در پوشه‌ی outs آمده است. همچنین برای اطمینان از درست ذخیره شدن هر کدام یک بار load شده و نشان داده شده است.

در بخش save models نیز هر مدل برای هر گوینده و با هر یک از ویژگی‌های MFCC و LPC با استفاده از pickle ذخیره شده است. این فایل‌ها با نام مناسب در پوشه‌ی outs آمده است. همچنین برای اطمینان از درست ذخیره شدن هر کدام یک بار load شده و نشان داده شده است.