

## سوال اول)

قطعه بندی به رویکردی گفته میشود که داده لای لایه میشود. به عنوان مثال در چشم پزشکی، با تشخیص لایه های چشم میتوانیم طبقه بندی بهتری داشته باشیم. اما در بحث داده های صوتی به صورت مختصر فواید زیر را دارد:

شناسایی واحدهای گفتاری: با قطعه بندی داده های صوتی، میتوان واحدهای کوچکتری از گفتار را مانند فریم های کوتاه زمانی (معمولاً 20-30 میلی ثانیه) شناسایی کرده و تحلیل کرد. این واحدهای گفتاری میتوانند به طور مفیدی برای شناسایی ویژگی های زبانی، مانند فونم ها، صدا های ترکیبی و تعیین موقعیت های توقف در گفتار مورد استفاده قرار گیرند.

با قطعه بندی داده های صوتی، میتوان ویژگی های مختلفی از گفتار را استخراج کرده و تحلیل کرد، از جمله ویژگی های مرتبط با فرکانس، محتوا، شدت و زمان. این ویژگی ها میتوانند به عنوان ورودی های مفید برای الگوریتم های تشخیص گفتار و یادگیری ماشین مورد استفاده قرار گیرند.

## سوال دوم)

### Mel Spectrogram

توضیح مختصر: طیف صوتی مل (Mel Spectrogram) نوعی از نمایش طیف صوتی است که از مقیاس مل برای توزیع فرکانس ها استفاده میکند، که به انسان اجازه میدهد فرکانس های مهم در طیف صوتی را با دقت بیشتری تشخیص دهد.

روابط: مل اسپکتروگرام به طور کلی از طیف صوتی محاسبه شده است. میتوان از آن برای استخراج ویژگی های دیگر مانند MFCC استفاده کرد.

### Zero Crossing Rate

توضیح مختصر: نرخ عبور صفر (Zero Crossing Rate) تعداد بارهایی است که سیگنال صوتی از محور صفر عبور میکند. این ویژگی معمولاً با سطح نویز و تنوع سیگنال صوتی مرتبط است.

روابط: نرخ عبور صفر میتواند به عنوان یکی از ویژگی های استخراج شده برای محاسبه سایر ویژگی ها مانند MFCC استفاده شود.

### MFCC (Mel-Frequency Cepstral Coefficients)

توضیح مختصر: ضرایب سبسترال فرکانس مل (Mel-Frequency Cepstral Coefficients) ویژگی های استخراج شده از طیف صوتی هستند که با استفاده از تبدیل مل فرکانس و تحلیل سبسترال به دست می آیند. این ویژگی ها معمولاً برای نمایش ویژگی های مهم گفتار استفاده میشوند.

روابط: MFCC معمولاً از مل اسپکتروگرام بدست می آیند و میتوانند با zero crossing rate و دیگر ویژگی های صوتی محاسبه شوند.

### Chroma Features

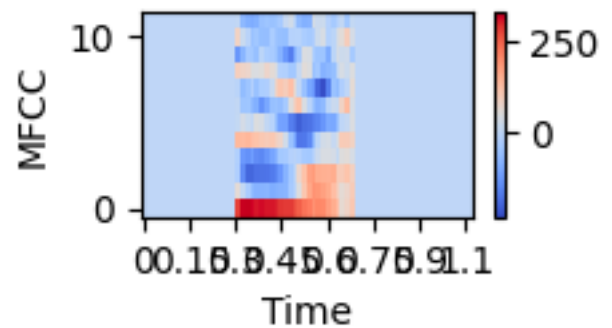
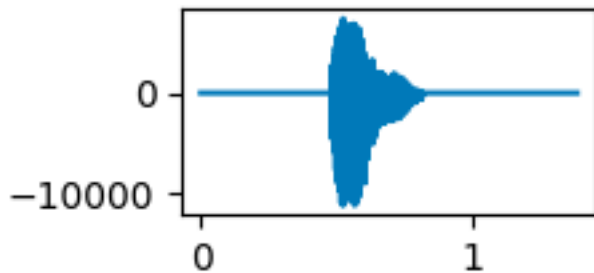
توضیح مختصر: ویژگی های کروما (Chroma Features) ویژگی هایی هستند که رنگ موسیقی را نمایان میکنند و به توزیع انرژی فرکانس های مختلف بر روی نت های موسیقی توجه میکنند.

روابط: Chroma features ممکن است از طریق تبدیل مل فرکانس و استخراج اطلاعات از آن ها، به ویژگی هایی مانند MFCC مرتبط باشند.

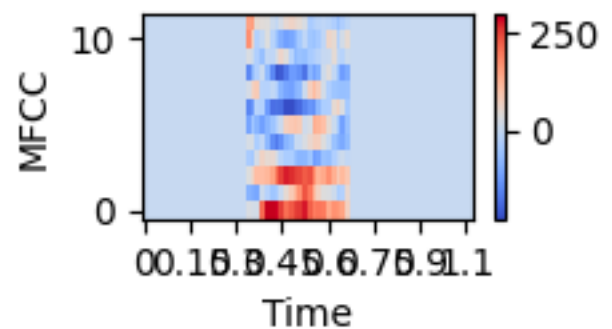
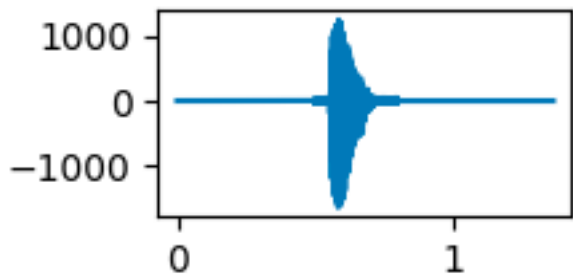
### سوال سوم)

ویژگی‌های MFCC به طور کلی به عنوان ویژگی‌های استاندارد برای تحلیل سیگنال‌های صوتی محسوب می‌شوند، زیرا به طور کلی مقاوم به تغییرات مختلف هستند و می‌توانند ویژگی‌های مهم گفتار را با دقت مناسبی استخراج کنند، حتی در مواجهه با مشکلات مختلف محیطی و صوتی. از این رو، MFCC به عنوان یک ویژگی robust در حوزه پردازش سیگنال‌های صوتی شناخته می‌شود.

5 sound audio wav



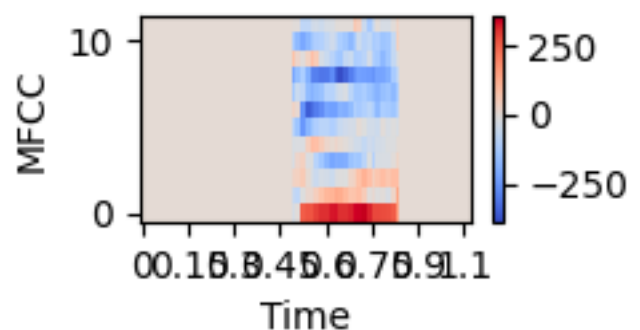
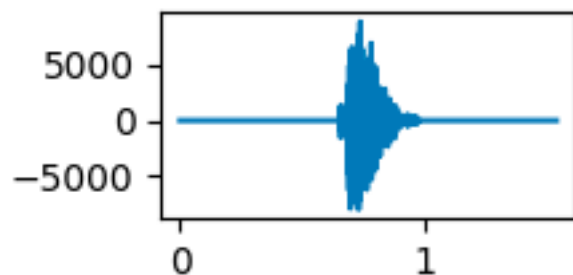
3 sound audio wav



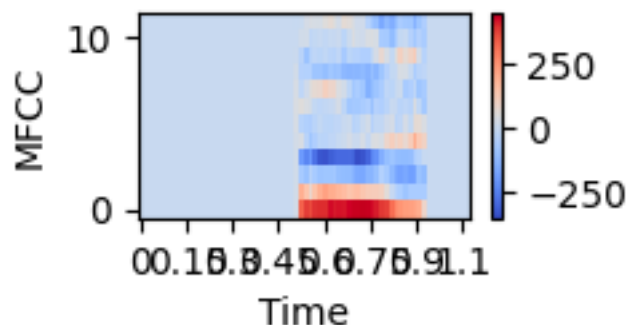
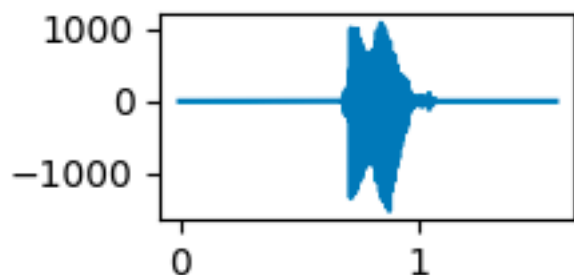
1 sound audio wav



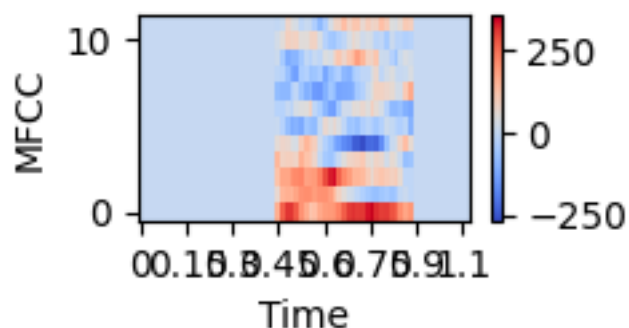
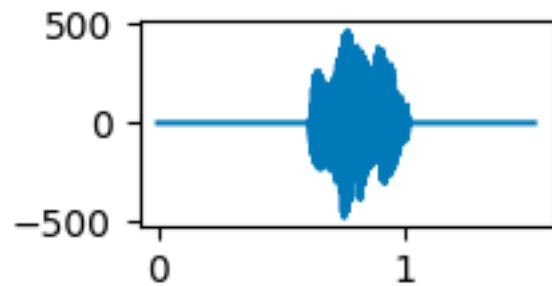
2 sound audio wav



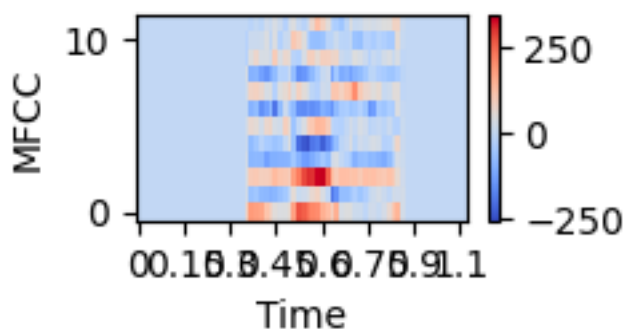
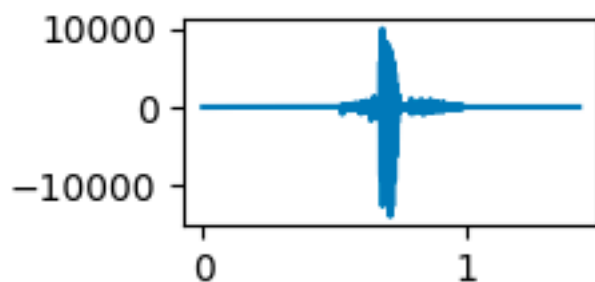
4 sound audio wav



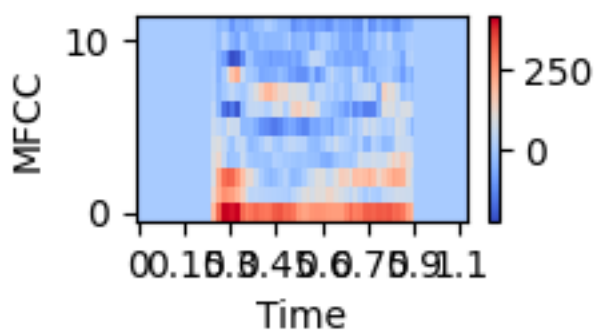
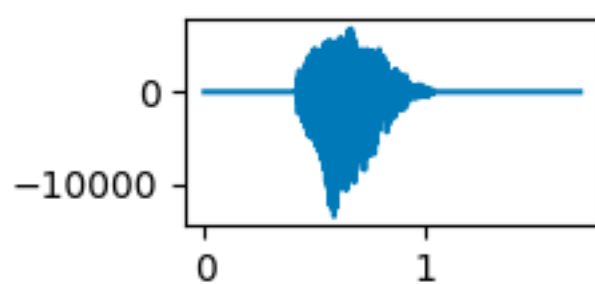
0 sound audio wav

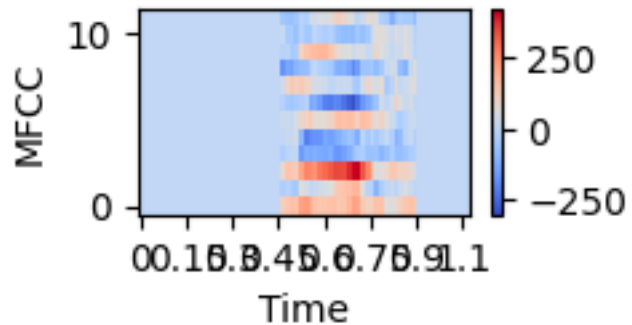
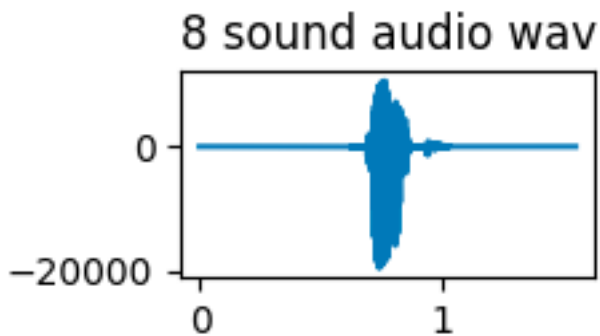
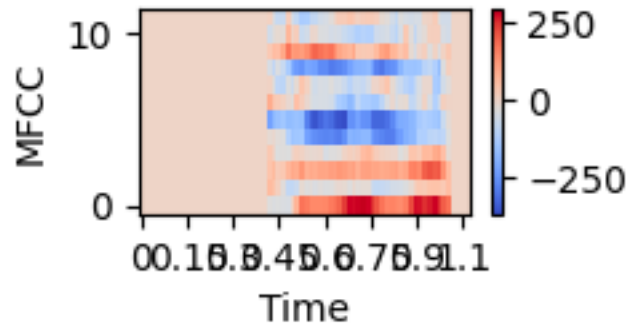
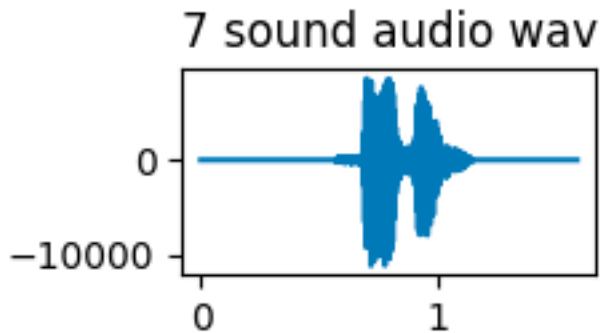


6 sound audio wav



9 sound audio wav





شکل ۱، نمونه ده تایی از mfcc های تولید شده

#### سوال چهارم)

نویز و تداخل: وجود نویز یا تداخل در سیگنال صوتی می‌تواند باعث افت کارایی MFCC شود، زیرا ممکن است فرکانس‌های مهم گفتار را پوشش دهد یا اطلاعات مربوط به فونم‌ها را از بین ببرد.

تغییرات در صدا: تغییرات در شدت، فرکانس و موقعیت صوت ممکن است باعث شود که معیارهای تبدیل و استخراج MFCC کارایی کمتری داشته باشند.

## سوال پنجم)

دو علت مهم:  
حفظ اطلاعات زمانی: با استفاده از همپوشانی بین فریم‌ها، می‌توانیم اطلاعات زمانی مهم در سیگنال را حفظ کنیم. زمانی که فریم‌ها با یکدیگر همپوشانی دارند، اطلاعات از تغییرات متوالی در سیگنال در طول زمان (مانند تغییرات در فونم‌ها، تغییرات در انرژی و ...) را نگه می‌دارند.  
کاهش افت کارایی در مرزها: هنگامی که فریم‌ها به صورت همپوشانی محاسبه می‌شوند، افت کارایی در مرزهای بین فریم‌ها کاهش می‌یابد. این افت کارایی معمولاً به دلیل اثر پنجره‌بندی (مانند پنجره همونگرایی) در سیگنال و ناتوانی در بازتاب صحیح اطلاعات در اطراف مرزها است.

## سوال ششم)

سرعت پردازش: با کاهش تعداد ضرایب MFCC، محاسبه و استفاده از این ویژگی‌ها سریع‌تر انجام می‌شود. این امر مهم است، به ویژه در برنامه‌هایی که نیاز به پردازش سیگنال‌های صوتی در زمان واقعی دارند.

کاهش ابعاد ویژگی: با استفاده از تنها ۱۲ یا ۱۳ ضریب ابتدایی MFCC، ابعاد ویژگی‌ها به طور قابل ملاحظه‌ای کاهش می‌یابد. این کاهش ابعاد معمولاً باعث بهبود کارایی مدل‌های یادگیری ماشینی می‌شود، زیرا اطلاعات زیادی که ممکن است غیرضروری باشند، حذف می‌شوند.

حفظ اطلاعات مهم: اکثر اطلاعات مهم مربوط به ویژگی‌های گفتاری در فرکانس‌های پایین تا متوسط قرار دارند. بنابراین، انتخاب ۱۲ یا ۱۳ ضریب ابتدایی MFCC به طور کلی کافی است برای نمایش ویژگی‌های اساسی و مهم گفتار.

## آشنایی با HMM

### سوال اول)

هر مشاهده یک ارایه دو بعدی یا یک mfcc میباشد. همچنین اینجا برای آموزش مدل به جای استفاده از ۱۰ استیت یک استیت در نظر گرفته شده است. یعنی استیت در هر مدل hmm دارای تنها یک رقم میباشد. البته اینکه استیت‌های پنهان در فریم‌های مختلف چقدر باشد یک پارامتر است که نتیجه نهایی را تغییر میدهد و قابل تغییر است. مثلاً اگر ۵ هیدن استیت در نظر بگیریم تمام آن ۵ هیدن استیت یک رقم هستند.  
ماتریس emission بعد از وارد کردن هر mfcc به تابع state likelihood محاسبه میشود.

### سوال دوم)

مدل HMM Order-First یک نوع از مدل‌های مخفی مارکوف (HMM) است که در آن توجه اصلی به انتقالات بین حالت‌ها در زمان فعلی وابسته است، به عبارت دیگر، تنها اولین درجه انتقالات را در نظر می‌گیرد. این مدل به دلیل این ویژگی به عنوان HMM Order-First نامگذاری می‌شود.

برای درک بهتر ویژگی‌های این مدل، ابتدا به مفهوم انتقالات در HMM اشاره می‌کنیم. در یک مدل HMM، فرض می‌شود که فرایندی وجود دارد که در آن یک سری از حالت‌های مخفی و یک سری از رخدادها (مشاهدات) وجود دارد. هر حالت مخفی به یک مشاهده خاص ارتباط دارد و با احتمال مشخصی به یک حالت جدید منتقل می‌شود. مدل HMM Order-First به طور خاص تنها از انتقالات به حالت‌های مجاور در زمان فعلی برای پیش‌بینی حالت بعدی استفاده می‌کند، به این معنی که احتمال انتقال به یک حالت جدید فقط از حالت فعلی مشتق می‌شود و احتمال انتقال به حالات دیگر در زمان فعلی نادیده گرفته می‌شود.

تفاوت اصلی این مدل با سایر مدل‌های HMM از جمله مدل HMM Order-Second و بالاتر، در تعداد درجه‌هایی است که در احتساب احتمالات انتقال به حالت‌های جدید مورد استفاده قرار می‌گیرد. در حالی که مدل‌های HMM با درجات بالاتر از اولین درجه، به عنوان مثال HMM Order-Second، از انتقالات به حالت‌های فرعی و دومین درجه نیز استفاده می‌کنند. این باعث می‌شود که این مدل‌ها برای مدل‌سازی وابستگی‌های زمانی پیچیده‌تر و متنوع‌تر بین حالت‌های مخفی عملکرد بهتری داشته باشند، اما به دلیل پیچیدگی بیشتر، ممکن است در مواردی که وابستگی‌های زمانی محدودتر هستند، استفاده از مدل‌های Order-First موثرتر و کارآمدتر باشد.

بنابراین، مدل HMM Order-First به عنوان یک روش ساده و موثر برای مدل‌سازی وابستگی‌های زمانی محدودتر بین حالت‌های مخفی در فرآیندهای زمانی مورد استفاده قرار می‌گیرد و به دلیل سادگی و کارایی آن، در بسیاری از حوزه‌ها از جمله پردازش سیگنال‌های صوتی، تشخیص گفتار، تحلیل زبان طبیعی، و ... مورد استفاده قرار می‌گیرد.

## سوال سوم)

مدل‌های مخفی مارکوف (Hidden Markov Models - HMMs) به عنوان یکی از مدل‌های آماری قدرتمند در حوزه‌های مختلفی از جمله پردازش سیگنال‌های صوتی، تشخیص گفتار، تحلیل زبان، بینایی ماشین، بیوانفورماتیک، رباتیک، و ... استفاده می‌شوند. این مدل‌ها به دلیل ویژگی‌های خاصی که دارند، برای بررسی و تحلیل پدیده‌هایی مختلف مناسب هستند:

پردازش سیگنال‌های صوتی و تشخیص گفتار HMMs: برای مدل‌سازی و تحلیل سیگنال‌های صوتی و پیش‌بینی واژگان و عبارات در تشخیص گفتار استفاده می‌شوند. این مدل‌ها می‌توانند وابستگی‌های زمانی در سیگنال‌های صوتی را مدل‌سازی کنند و با استفاده از آن‌ها می‌توان تشخیص داد که چه کلمات و عباراتی در یک سیگنال صوتی وجود دارند.

تحلیل زبان و متن HMMs: می‌توانند برای تحلیل زبان و متن استفاده شوند، به عنوان مثال برای مدل‌سازی نحوه تولید جملات در زبان یا تشخیص ترتیب و نحوه ظاهر شدن کلمات در متون.

بینایی ماشین و تشخیص الگو: در حوزه بینایی ماشین، HMMs برای تشخیص الگو و شناسایی اشیاء در تصاویر مورد استفاده قرار می‌گیرند. به عنوان مثال، برای تشخیص حرکت یک شی در یک تصویر ویدیویی.

پردازش زبان طبیعی: در حوزه پردازش زبان طبیعی، HMMs می‌توانند برای مدل‌سازی زبان، تحلیل و تفسیر ساختار جملات، ترجمه ماشینی و ... استفاده شوند.

بیوانفورماتیک: در بیوانفورماتیک، HMMs برای مدل‌سازی و تحلیل ساختار پروتئین‌ها، پیش‌بینی ساختار سه بعدی پروتئین‌ها، تحلیل سیگنال‌های ژنتیکی، و ... استفاده می‌شوند.

چرا که مدل‌های HMMs قابلیت مدل‌سازی وابستگی‌های زمانی و نحوه تغییرات در زمان را دارند، این مدل‌ها به خوبی برای بررسی و تحلیل پدیده‌هایی که وابستگی به تغییرات در زمان دارند، مناسب هستند. به عنوان مثال، در تشخیص گفتار، واژگان و عبارات به ترتیب زمانی خاصی ظاهر می‌شوند که با استفاده از مدل‌های HMMs می‌توان این وابستگی‌های زمانی را مدل‌سازی کرد و به تشخیص گفتار پرداخت. به همین دلیل، این مدل‌ها در بسیاری از حوزه‌های تحلیل و پردازش داده که وابستگی به زمان دارند، کارآمد هستند.

## سوال چهارم)

مزایا:

قابلیت مدل سازی وابستگی های زمانی HMMs: قادر به مدل سازی وابستگی های زمانی در داده ها هستند. این به این معنی است که می توانند تغییرات زمانی در داده را به خوبی درک و مدل کنند، که بسیار مفید است برای بررسی سیگنال های زمانی مثل سیگنال های صوتی و ویدیویی.

ارزشمند برای داده های پویا HMMs: بسیار مناسب برای داده های پویا و متغیر در زمان هستند، زیرا این مدل ها می توانند وضعیت های مختلفی که یک داده می تواند در زمان از خود نشان دهد را مدل کنند.

ساختار گرافی ساده: ساختار گرافی مدل HMM نسبتاً ساده است که موجب می شود محاسبات با این مدل ها به راحتی انجام شود و برای آموزش و پیش بینی ها مناسب باشد.

معایب:

فرضیات ساده سازی: یکی از مشکلات اصلی مدل های HMM این است که فرضیات ساده سازی شده ای راجع به داده را دارند، از جمله فرضیات مستقلی بودن داده در زمان های مختلف و توزیع های گوسی بودن داده.

مشکل در مدل سازی پدیده های پیچیده HMMs: به خوبی با پدیده هایی که وابستگی های زمانی پیچیده ای دارند، کنار نمی آیند. بنابراین، برای مدل سازی پدیده های پیچیده تر ممکن است نیاز به استفاده از مدل های پیشرفته تری باشد.

حساسیت به شرایط آموزشی HMMs: حساس به شرایط آموزشی هستند و ممکن است در صورت کمبود داده یا شرایط آموزش نامناسب، عملکرد ناپایداری داشته باشند.

پیدا کردن تعداد مناسب حالت های مخفی: انتخاب تعداد مناسبی از حالت های مخفی در مدل HMM می تواند مسأله ای دشوار باشد و نیاز به تخمین دقیق و آزمایش و خطا دارد.

پیدا کردن مسیر مخفی بهینه: در برخی از موارد، پیدا کردن مسیر مخفی بهینه در مدل HMM ممکن است مسأله ای محاسباتی پیچیده باشد، به خصوص در مدل های با تعداد حالت های مخفی بالا و داده های بزرگ.

## سوال پنجم)

HMM گوسی (Gaussian HMM)

در این نوع از مدل های HMM، فرض می شود که توزیع های احتمال برای مشاهدات در هر حالت مخفی از توزیع گوسی پیروی می کنند.

این نوع از مدل ها معمولاً برای داده های پیوسته استفاده می شوند که با استفاده از توزیع گوسی می توانند توزیع احتمالات مشاهدات را مدل کنند.

مناسب برای مسائلی که مشاهدات به صورت پیوسته هستند مانند پردازش سیگنال های صوتی و تصویری است.

HMM دسته بندی شده (Multinomial HMM)

در این نوع از مدل های HMM، توزیع احتمالات برای مشاهدات در هر حالت مخفی به صورت یک توزیع چند جمله ای مدل می شود.

هوش مصنوعی  
قسمت عملی  
تمرین دوم  
ریحانه اسماعیلزاده  
۸۱۰۸۰۰۰۰۴

این نوع از مدل‌ها معمولاً برای داده‌های گسسته و کاتگوریکال استفاده می‌شوند که می‌توانند با استفاده از توزیع چندجمله‌ای مدل شوند.

مورد استفاده در بسیاری از مسائل دسته‌بندی و تصمیم‌گیری مانند تحلیل متن، شناسایی الگوها، و ... است.  
HMM مخلوط گوسی (Gaussian Mixture HMM)

این نوع از مدل‌های HMM، ترکیبی از چندین توزیع گوسی را برای مدل‌سازی توزیع احتمالات مشاهدات در هر حالت مخفی استفاده می‌کنند.

این نوع از مدل‌ها قابلیت مدل‌سازی توزیعات پیچیده‌تر را نسبت به HMM گوسی ارائه می‌دهند و بهتر می‌توانند داده‌هایی با توزیع‌های غیر گوسی را نیز مدل کنند.

معمولاً برای مسائلی که توزیعات داده‌ها پیچیده و غیر گوسی هستند مورد استفاده قرار می‌گیرند.  
HMM مخلوط چندجمله‌ای (Multinomial Mixture HMM)

این نوع از مدل‌های HMM، ترکیبی از چندین توزیع چندجمله‌ای را برای مدل‌سازی توزیع احتمالات مشاهدات در هر حالت مخفی استفاده می‌کنند.

مانند HMM مخلوط گوسی، این نوع از مدل‌ها بهتر می‌توانند توزیعات پیچیده‌تر و غیر گوسی را نیز مدل کنند.  
HMM زمانی (Temporal HMM)

در این نوع از مدل‌های HMM، وابستگی زمانی بین حالت‌های مخفی مدل شده است و اطلاعات زمانی در فرآیند تصمیم‌گیری در نظر گرفته می‌شود.  
این نوع از مدل‌ها برای مسائلی که دارای دینامیک زمانی هستند و وابستگی‌های زمانی مهمی در داده‌ها وجود دارد، مناسب هستند.

## استفاده از کتابخانه آماده برای آموزش hmm

### پیش پردازش داده ها

#### ۱. حذف سکوت از صداها

```
from pydub.silence import split_on_silence
from pydub import AudioSegment

def remove_silence(input_path, output_path, min_silence_len = 50, silence_thresh = -40):
    audio = AudioSegment.from_mp3(input_path)

    chunks = split_on_silence(audio, min_silence_len = min_silence_len, silence_thresh = silence_thresh)
    output = AudioSegment.empty()
    for chunk in chunks:
        output += chunk

    output.export(output_path, format = "wav")
```

یکی از مهم ترین پیش پردازش های ما حذف سکوت میباشد.



## ۲. نرمالیز کردن داده ها به روش min max

```
from sklearn.model_selection import train_test_split
from collections import defaultdict

def normalise(array):
    min_val = 0
    max_val = 1
    norm_array = (array - array.min()) / (array.max() - array.min())
    norm_array = norm_array * (max_val - min_val) + min_val
    return norm_array

# Create dictionaries to store train and test MFCC features for each digit label
train_mfcc_by_digit = defaultdict(list)
test_mfcc_by_digit = defaultdict(list)
# Split MFCC features for each digit into train and test sets
for label, mfcc_list in mfcc_by_digit.items():
    train_mfcc, test_mfcc = train_test_split(mfcc_list, test_size=0.2,
random_state=42)

    # Normalize train MFCCs
    train_mfcc = [normalise(mfcc) for mfcc in train_mfcc]
    train_mfcc_by_digit[label].extend(train_mfcc)

    # Normalize test MFCCs
    test_mfcc = [normalise(mfcc) for mfcc in test_mfcc]
    test_mfcc_by_digit[label].extend(test_mfcc)

# Create dictionaries to store train and test MFCC features for each digit label
train_mfcc_by_speaker = defaultdict(list)
test_mfcc_by_speaker = defaultdict(list)
# Split MFCC features for each digit into train and test sets
for speaker, mfcc_list in mfcc_by_speaker.items():
    train_mfcc_speaker, test_mfcc_speaker = train_test_split(mfcc_list, test_size=0.2,
random_state=42)

    # Normalize train MFCCs
    train_mfcc_speaker = [normalise(mfcc) for mfcc in train_mfcc]
    train_mfcc_by_speaker[speaker].extend(train_mfcc_speaker)

    # Normalize test MFCCs
    test_mfcc_speaker = [normalise(mfcc) for mfcc in test_mfcc_speaker]
    test_mfcc_by_speaker[speaker].extend(test_mfcc_speaker)
```

هوش مصنوعی  
قسمت عملی  
تمرین دوم  
ریحانه اسماعیلیزاده  
۸۱۰۸۰۰۰۰۴

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

نرمالیز کردن داده ها به روش *min max*

تمرین داده ها با مدل Gaussian HMM

```
from hmmlearn import hmm
from collections import defaultdict
import numpy as np

# Assuming you have filtered_train_mfcc_by_digit containing filtered MFCCs for each
digit

# Define the number of states for the HMM models
num_states = 4 # You can adjust this as needed

# Dictionary to store trained HMM models for each digit
hmm_models = {}

# Train HMM models for each digit
for digit, mfcc_list in filtered_train_mfcc_by_digit.items():
    # Concatenate filtered MFCCs for this digit into a single array
    digit_mfccs = np.concatenate(mfcc_list, axis=1) # Concatenate along the second
axis (frames)

    # Create and train HMM model
    model = hmm.GaussianHMM(n_components=num_states)
    model.fit(digit_mfccs.T) # Transpose the array to match hmmlearn's convention

    # Store the trained model
    hmm_models[digit] = model
```

```
correct_predictions = 0
total_predictions = 0

# Iterate through each test sample
for true_digit, mfcc_list in filtered_test_mfcc_by_digit.items():
```

هوش مصنوعی  
قسمت عملی  
تمرین دوم  
ریحانه اسماعیلیزاده  
۸۱۰۸۰۰۰۰۴

```
for mfcc in mfcc_list:
    # Compute log-likelihood under each trained HMM model
    likelihoods = {digit: model.score(mfcc.T) for digit, model in
hmm_models.items()}

    # Assign the sample to the digit with the highest log-likelihood
    predicted_digit = max(likelihoods, key=likelihoods.get)

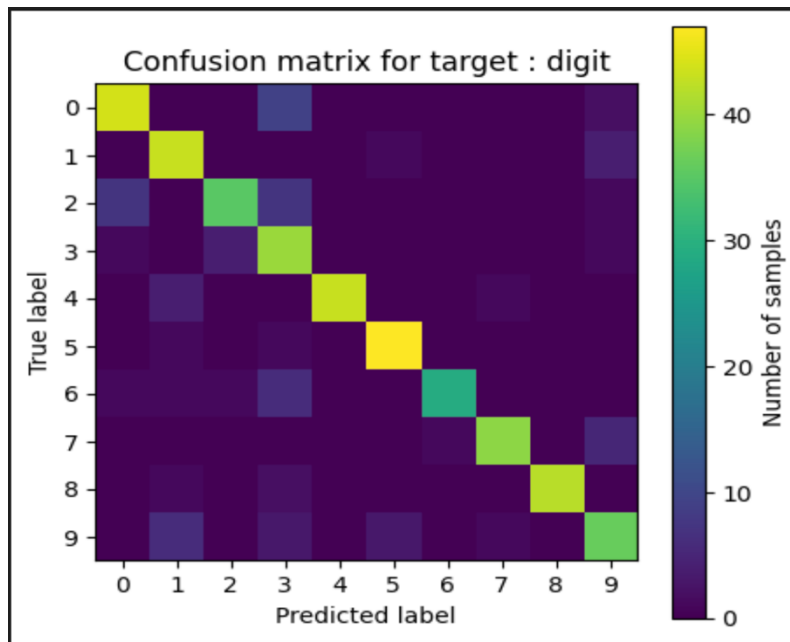
    # Check if prediction is correct
    if predicted_digit == true_digit:
        correct_predictions += 1
    total_predictions += 1

# Calculate accuracy
accuracy = correct_predictions / total_predictions
print("Accuracy:", accuracy)
```

علت استفاده از Gaussian HMM این است که ما می‌خواهیم مقادیر موجود در emission matrix را با normal distribution حساب کنیم.

Accuracy: 0.8414376321353065

دقت به تنهایی دلیل کافی بر اثبات درست بودن مدل نیست. برای اطمینان باید Confusion Matrix رسم شود:



هوش مصنوعی  
قسمت عملی  
تمرین دوم  
ریحانه اسماعیلزاده  
۸۱۰۸۰۰۰۰۴

همانطور که مشاهده میشود و طبق انتظار ما، تعداد داده های روی قطر ماتریس به سمت رنگ گرم رفته اند که یعنی هر عدد به رقم درست پیش بینی شده است.

معیار های دیگر ارزیابی:

Precision and Recall and F1 score

$$\text{F1 Score} = \frac{2}{\left(\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}\right)}$$

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

		Predicted	
		0	1
Actual	0	TN	FP
	1	FN	TP

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

## سوال اول و سوم

**دقت (Precision):** دقت نشان دهنده این است که از مواردی که مدل به عنوان مثبت شناخته است، چه تعدادی واقعا مثبت هستند.

- ☐ **TP** تعداد مواردی است که به درستی تشخیص داده شده‌اند. (True Positives)
  - ☐ **FP** تعداد مواردی است که به اشتباه به عنوان مثبت تشخیص داده شده‌اند. (False Positives)
- اگر دقت بالا باشد، این نشان می‌دهد که مدل به خوبی می‌تواند مثبت‌ها را تشخیص دهد و از اشتباهات کمتری رنج می‌برد.

**بازیابی (Recall):** بازیابی نشان می‌دهد که مدل چه تعداد از موارد مثبت واقعی را تشخیص داده است.

- ☐ **FN** تعداد مواردی است که به اشتباه به عنوان منفی تشخیص داده شده‌اند. (False Negatives)

اگر بازیابی بالا باشد، این نشان می‌دهد که مدل بسیاری از موارد مثبت را پیدا کرده است و به طور کلی از دقت خوبی برخوردار است.

**امتیاز (F1 Score):** امتیاز F1 میانگین هارمونیک دقت و بازیابی است و از آن به عنوان یک معیار جامع برای ارزیابی عملکرد یک مدل استفاده می‌شود.

امتیاز F1 به طور خاص مفید است زیرا هر دو دقت و بازیابی را در نظر می‌گیرد و در مواقعی که تعداد مثبت‌ها و منفی‌ها ناهمسان است، عملکرد بهتری نسبت به معیارهای دیگر ارائه می‌دهد.

- ☐ اگر دقت بالا و بازیابی پایین باشد، این نشان می‌دهد که مدل به مثبت‌ها بسیار حساس است ولی ممکن است بسیاری از مثبت‌ها را از دست دهد. به عبارت دیگر، مدل به اشتباهات کمتری دچار می‌شود، اما می‌تواند مثبت‌های واقعی را از دست بدهد.
  - ☐ اگر دقت پایین و بازیابی بالا باشد، این نشان می‌دهد که مدل بیشتر از مثبت‌های واقعی را شناسایی کرده است، اما احتمال اشتباه در تشخیص مثبت‌ها بسیار بیشتر است.
  - ☐ اگر هر دو دقت و بازیابی بالا باشد، این نشان دهنده یک مدل خوب و کارآمد است که هم به خوبی مثبت‌ها را شناسایی می‌کند و هم از اشتباهات کمی رنج می‌برد.
  - ☐ اگر هر دو دقت و بازیابی پایین باشد، این نشان دهنده عدم کارایی مدل است و نشان می‌دهد که مدل نه تنها مثبت‌ها را به درستی شناسایی نمی‌کند بلکه اشتباهات زیادی هم دارد.
- با توجه به متناسب بودن دقت و بازیابی در معیار F1، زمانی که به دنبال معیار جامعی برای ارزیابی مدل هستید، امتیاز F1 را مورد استفاده قرار دهید.
- همه سه معیار (دقت، بازیابی، و امتیاز F1) بین ۰ و ۱ متغیرند.

Average Precision: 0.8635429800083863			
Average Recall: 0.8418267329815222			
Average F1-score: 0.8466454040122493			
Class	Precision	Recall	F1-score
0	0.8302	0.8000	0.8148
1	0.7679	0.8958	0.8269
2	0.8750	0.7000	0.7778
3	0.5882	0.8696	0.7018
4	1.0000	0.8958	0.9451
5	0.9216	0.9592	0.9400
6	0.9667	0.7632	0.8529
7	0.9512	0.8667	0.9070
8	1.0000	0.9333	0.9655
9	0.7347	0.7347	0.7347

F1 score برای اکثریت این کلاس‌ها بالا است و به معنی خوب بودن مدل در پیش‌بینی می‌باشد.

## سوال دوم)

بله

نویز در داده‌ها

حضور نویز در داده‌ها می‌تواند تاثیر بسیار زیادی بر روی معیارهای ارزیابی داشته باشد. برخی از مدل‌ها حساسیت بالا به داده‌های نویزی دارند که ممکن است باعث کاهش عملکرد مدل و افزایش اشتباهات در تخمین معیارها شود.

تعداد داده‌های ناپایدار برای هر کلاس:

ممکن است تعداد داده‌های موجود برای هر کلاس متفاوت باشد. این موضوع می‌تواند به عدم توازن کلاس‌ها و انحراف در ارزیابی معیارها منجر شود. برای مثال، اگر یک کلاس دارای تعداد داده‌های بسیار کمی باشد، معیارها ممکن است دقت و صحت بالایی نشان دهند اما این وضعیت نمایانگر عملکرد واقعی مدل نیست.

مقیاس داده‌ها:

ممکن است داده‌ها مقیاس‌های مختلفی داشته باشند که می‌تواند به دقت ارزیابی مدل و معیارهای دقت، بازیابی و امتیاز F1 تاثیر بگذارد. در برخی مواقع، نیاز به استانداردسازی داده‌ها برای حل این مشکل و ارزیابی صحیح مدل وجود دارد.

## سوال چهارم)

(Recall بازیابی) نشان دهنده توانایی مدل در شناسایی تمام نمونه‌های مثبت در دسته مورد نظر است. به عبارت دیگر، بازیابی نشان می‌دهد که مدل چه تعداد از کل نمونه‌های مثبت را به درستی شناسایی کرده است.

(Precision دقت) نشان دهنده توانایی مدل در تشخیص مثبت‌ها است و نسبت مثبت‌های درست تشخیص داده شده توسط مدل به تعداد کل مثبت‌هایی است که مدل تشخیص داده است.

چرا هر کدام به تنهایی برای ارزیابی مدل کافی نیست؟

برای: Precision

اگر Precision بالا باشد، این نشان می‌دهد که مدل تمایل دارد که وقتی یک مثبت تشخیص می‌دهد، واقعاً مثبت باشد. اما اگر مدل بسیار دقیق باشد اما توانایی شناسایی همه مثبت‌ها را نداشته باشد (Recall پایین باشد)، ممکن است مدل اطلاعات مهم را از دست بدهد و نتواند یک تصمیم صحیح بگیرد.

برای: Recall

اگر Recall بالا باشد، این نشان می‌دهد که مدل تمایل دارد تا تمام مثبت‌ها را شناسایی کند. اما اگر مدل به صورت تصادفی مثبت‌ها را تشخیص دهد (Precision پایین باشد)، ممکن است بیشتر مثبت‌های تشخیص داده شده اشتباه باشند و عملکرد مدل اعتباری نداشته باشد.

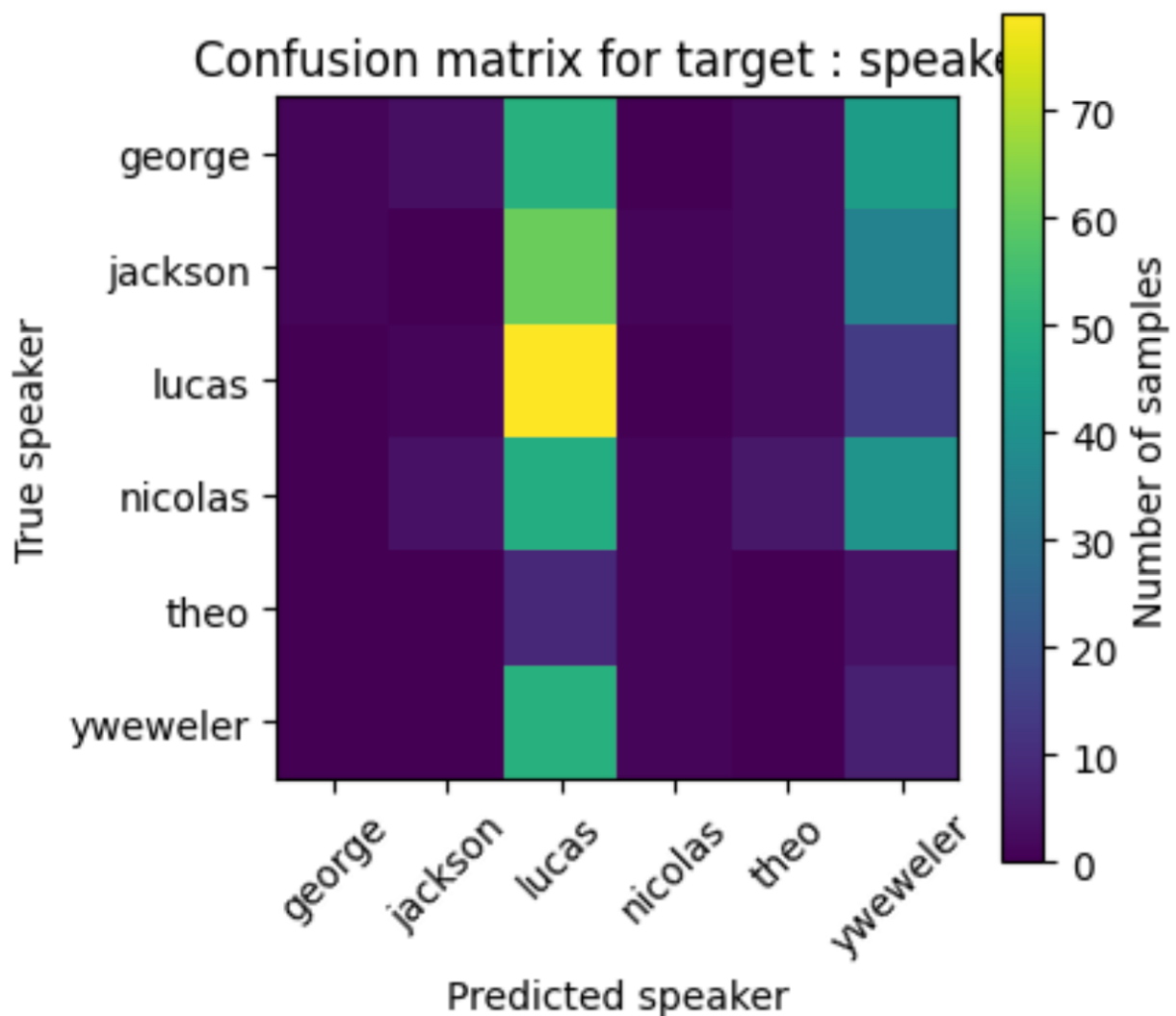
مثال:

فرض کنید یک مدل داریم که برای مجموعه داده ای با دو کلاس، A و B آموزش داده شده است. اگر مدل با یک آستانه بالا به اندازه کافی دقیق باشد تا تمامی نمونه‌های مثبت را به درستی تشخیص دهد (Recall بالا)، اما به طور ناپذیری مثبت‌های بیشتری را نیز به عنوان مثبت تشخیص دهد (Precision پایین)، این مدل به ظاهر با Recall بالا به نظر موفق می‌آید، اما در واقعیت به دلیل دقت پایین، اطلاعات معتبری ارائه نمی‌دهد و نمی‌تواند به صورت قابل اعتمادی مثبت‌ها را تشخیص دهد.

### سوال پنجم)

معیار F1 از میانگین هارمونیک استفاده می‌کند. میانگین هارمونیک در واقع برعکس میانگین اعداد است. تفاوت میانگین هارمونیک با میانگین عادی این است که در میانگین هارمونیک، ارزش کوچک‌ترین عدد موثر است. این به این معنی است که اگر یکی از مقادیر صفر باشد، میانگین هارمونیک نیز صفر خواهد بود. اهمیت استفاده از میانگین هارمونیک در معیار F1 این است که این معیار به دنبال تعادل بین دقت (Precision) و بازیابی (Recall) است. چرا که اگر یکی از این دو معیار بسیار بالا و دیگری بسیار پایین باشد، میانگین هندسی یا معمولی احتمالاً به شدت تحت تأثیر قرار می‌گیرد. از طرف دیگر، میانگین هارمونیک باعث می‌شود که مقادیر پایین‌تر بیشتر تأثیر بگذارند و از نتیجه‌ی ناکارآمدی در صورت وجود ارزش‌های پایین‌تر در یکی از معیارها، جلوگیری می‌کند.

### عوض کردن هدف به گوینده



```
Average Precision: 0.17722942220165086
Average Recall: 0.16060105363984675
Average F1-score: 0.08480322633929989

Speaker      Precision      Recall      F1-score
george        0.5000        0.0100      0.0196
jackson       0.0000        0.0000      0.0000
lucas         0.2651        0.8229      0.4010
nicolas       0.2500        0.0100      0.0192
theo          0.0000        0.0000      0.0000
yweweler      0.0483        0.1207      0.0690
```

- ☐ Speaker george: برای  
دقت: 50.00%  
بازیابی: 1.00%  
امتیاز F1: 1.96%  
عملکرد نه چندان خوب برای Speaker george ، زیرا دقت پایین و بازیابی بسیار پایین است.
- ☐ Speaker jackson:  
دقت: 0.00%  
بازیابی: 0.00%  
امتیاز F1: 0.00%  
عملکرد بسیار ضعیف برای Speaker jackson ، زیرا هیچ تشخیص درستی صورت نگرفته است.
- ☐ Speaker lucas:  
دقت: 26.51%  
بازیابی: 82.29%  
امتیاز F1: 40.10%  
عملکرد متوسط برای Speaker lucas ، با دقت کم و بازیابی قابل قبول.
- ☐ Speaker nicolas:  
دقت: 25.00%  
بازیابی: 1.00%  
امتیاز F1: 1.92%  
عملکرد نه چندان خوب برای Speaker nicolas ، با دقت پایین و بازیابی بسیار پایین.
- ☐ Speaker theo:



هوش مصنوعی  
قسمت عملی  
تمرین دوم  
ریحانه اسماعیلزاده  
۸۱۰۸۰۰۰۰۴

- ☐ دقت: 0.00%
- ☐ بازیابی: 0.00%
- ☐ امتیاز F1: 0.00%
- ☐ عملکرد بسیار ضعیف برای Speaker theo ، زیرا هیچ تشخیص درستی صورت نگرفته است.
- ☐ Speaker yweweler:
- ☐ دقت: 4.83%
- ☐ بازیابی: 12.07%
- ☐ امتیاز F1: 6.90%
- ☐ عملکرد ضعیف برای Speaker yweweler ، با دقت و بازیابی پایین.

به طور کلی، مدل نتوانسته است به خوبی صحبت‌کنندگان را شناسایی کند، زیرا دقت و بازیابی برای اکثریت صحبت‌کنندگان پایین است و امتیاز F1 هم بسیار پایین است. این نشان می‌دهد که مدل بهبود بیشتری نیاز دارد تا به یک عملکرد قابل قبول برسد.

1. **تفاوت در تنوع کلاس‌ها:** تعداد و تنوع کلاس‌ها (speaker) ممکن است از تعداد و تنوع کلاس‌های اعداد متفاوت باشد. این تغییر ممکن است باعث شود که مدل با کارایی کمتری عمل کند، به خصوص اگر تعداد نمونه‌ها برای هر کلاس یکسان نباشد.
2. **تفاوت در ویژگی‌ها:** ممکن است ویژگی‌هایی که برای تشخیص اعداد مورد استفاده قرار گرفته بود، برای تشخیص صحبت‌کنندگان مناسب نباشد. برای تشخیص صحبت‌کنندگان، ممکن است نیاز به ویژگی‌های دیگر یا ترکیب‌های مختلفی از ویژگی‌ها باشد که مدل ما آنها را در نظر نگرفته است.

## استفاده از کد from scratch برای آموزش hmm

### برای شناسایی digits

عملیات پیش پردازش همانند مرحله قبل انجام میشود  
منتهی در اینجا به شیوه جدیدی به حذف های ویس های خالی میپردازیم.

```
if audio.size == 0:
    print(f"Skipping short or empty audio file: {x_train_paths[j]}")
    continue
# Pad the audio signal with zeros to make it at least min_audio_length long
if len(audio) < min_audio_length:
    audio = np.pad(audio, (0, min_audio_length - len(audio)), 'constant')
coefs = mfcc(audio, sampling_freq, nfft=1024)
```

Average Precision: 0.800412034645387  
Average Recall: 0.7549453286935498  
Average F1-score: 0.7410890449982314  
Class Precision Recall F1-score

0	0.7432	0.8871	0.8088
1	0.6250	0.9559	0.7558
2	0.9104	0.8714	0.8905
3	0.6923	0.4091	0.5143
4	0.9130	0.6562	0.7636
5	0.9808	0.7969	0.8793
6	0.5952	0.8065	0.6849
7	0.6263	0.9394	0.7515
8	0.9178	0.9571	0.9371
9	1.0000	0.2698	0.4250

### ارزیابی مدل با این معیارها

#### Speaker 0

- ☐ دقت: 74.32%
- ☐ بازیابی: 88.71%
- ☐ امتیاز F1: 80.88%
- ☐ عملکرد خوبی در تشخیص Speaker 0 داریم، اما می‌توانیم سعی کنیم بازیابی را افزایش دهیم تا بهترین عملکرد را داشته باشیم.

#### Speaker 1

- ☐ دقت: 62.50%
- ☐ بازیابی: 95.59%
- ☐ امتیاز F1: 75.58%
- ☐ دقت متوسطی داریم، اما بازیابی بسیار بالاست که نشان می‌دهد مدل به خوبی قادر به شناسایی Speaker 1 است.

#### Speaker 2

- ☐ دقت: 91.04%
- ☐ بازیابی: 87.14%
- ☐ امتیاز F1: 89.05%
- ☐ عملکرد خوبی در تشخیص Speaker 2 داریم، با دقت و بازیابی بالا و امتیاز F1 قابل قبول.

#### Speaker 3

- ☐ دقت: 69.23%
- ☐ بازیابی: 40.91%
- ☐ امتیاز F1: 51.43%
- ☐ دقت مناسبی داریم، اما بازیابی کم است که نشان می‌دهد مدل نیاز به بهبود در تشخیص Speaker 3 دارد.

#### Speaker 4

هوش مصنوعی  
قسمت عملی  
تمرین دوم  
ریحانه اسماعیلزاده  
۸۱۰۸۰۰۰۰۴

دقت: 91.30% ☐

بازیابی: 65.62% ☐

امتیاز F1: 76.36% ☐

دقت خوبی داریم، اما بازیابی متوسط است که ممکن است نیاز به بهبود داشته باشد.

#### Speaker 5

دقت: 98.08% ☐

بازیابی: 79.69% ☐

امتیاز F1: 87.93% ☐

دقت بسیار بالا بازیابی متوسط و امتیاز F1 بالاست که نشان از خوبی عملکرد مدل در تشخیص

Speaker 5 دارد.

#### Speaker 6

دقت: 59.52% ☐

بازیابی: 80.65% ☐

امتیاز F1: 68.49% ☐

دقت پایینی داریم اما بازیابی مناسبی که نشان می‌دهد مدل قادر به شناسایی Speaker 6 است.

#### Speaker 7

دقت: 62.63% ☐

بازیابی: 93.94% ☐

امتیاز F1: 75.15% ☐

دقت متوسط با بازیابی بالا که نشان می‌دهد مدل به خوبی قادر به شناسایی Speaker 7 است.

#### Speaker 8

دقت: 91.78% ☐

بازیابی: 95.71% ☐

امتیاز F1: 93.71% ☐

عملکرد بسیار بالا با دقت و بازیابی بالا که نشان می‌دهد مدل بهترین عملکرد را در تشخیص Speaker 8 دارد.

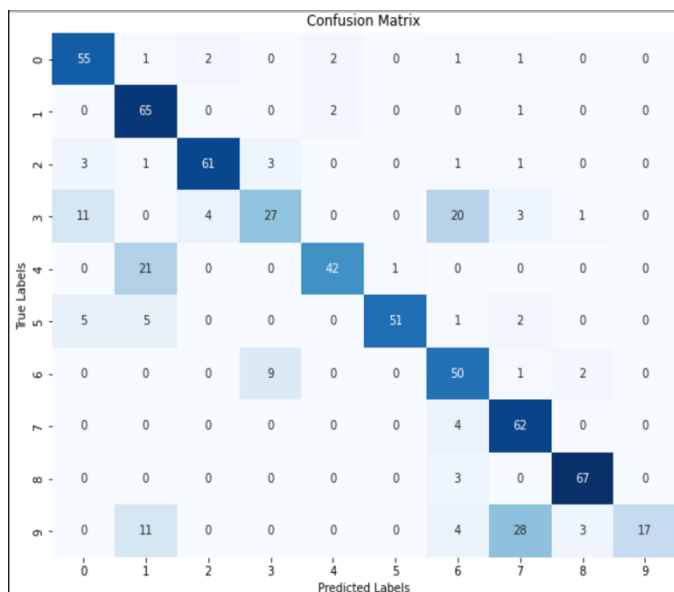
#### Speaker 9

دقت: 100.00% ☐

بازیابی: 26.98% ☐

امتیاز F1: 42.50% ☐

دقت بسیار بالا اما بازیابی پایین که نشان می‌دهد مدل در تشخیص Speaker 9 به عملکرد مناسبی نمی‌رسد.



Con matrix نیز خوب بودن مدل در پیش بینی را نشان میدهد.  
تغییر به هدف گوینده

```
accuracy is : 0.707395498392283
Average Precision: 0.7859301719645649
Average Recall: 0.7067147540983607
Average F1-score: 0.6863708972787101
```

Class	Precision	Recall	F1-score
0	0.5895	0.8960	0.7111
1	1.0000	0.2800	0.4375
2	1.0000	0.8000	0.8889
3	0.6139	0.9920	0.7584
4	0.7263	0.5656	0.6359

جالب توجه است تمام فایل های theoxراب بوده و تنها ۵ مورد مورد استفاده قرار گرفت.

#### کلاس 1:

- ☐ دقت: 1.0000
- ☐ بازیابی: 0.2800
- ☐ امتیاز F1: 0.4375
- ☐ دقت بسیار بالا و بازیابی پایین نشان می‌دهند که مدل تمایل به تشخیص کلاس 1 را دارد، اما به اشتباه تشخیص داده‌های دیگر را به این کلاس اختصاص می‌دهد. امتیاز F1 معمولاً میانگین هندسی این دو معیار است.

#### کلاس 2:

- ☐ دقت: 1.0000
- ☐ بازیابی: 0.8000
- ☐ امتیاز F1: 0.8889
- ☐ دقت بالا و بازیابی مناسب نشان می‌دهند که مدل توانایی خوبی در تشخیص کلاس 2 دارد. امتیاز F1 نیز بسیار بالاست.

#### کلاس 3:

- ☐ دقت: 0.6139

بازیابی: 0.9920 ☐

امتیاز F1: 0.7584 ☐

در اینجا، بازیابی بسیار بالا و دقت نسبتاً متوسط نشان می‌دهد که مدل تمایل دارد تا به درستی داده‌های این کلاس را تشخیص دهد، اما ممکن است به اشتباه داده‌های دیگری را به این کلاس اختصاص دهد.

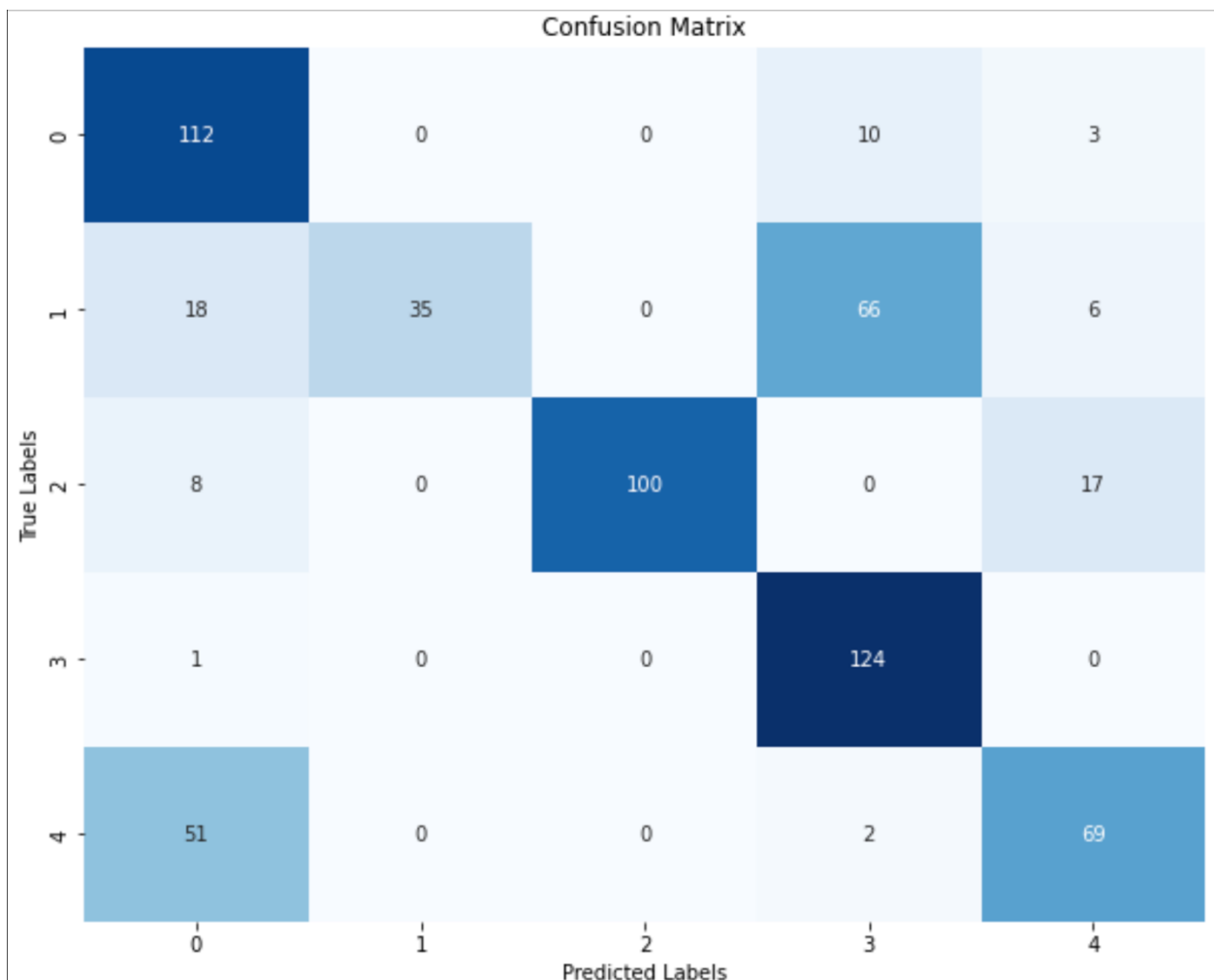
#### کلاس 4:

دقت: 0.7263 ☐

بازیابی: 0.5656 ☐

امتیاز F1: 0.6359 ☐

دقت و بازیابی متوسطی که نشان می‌دهد که مدل در تشخیص کلاس 4، به طور متوسط عمل می‌کند و امتیاز F1 نیز متناسب با این دو معیار است.



همانطور که پیداست گوینده ها توانسته اند با دقت ۷۰ درصد تخمین زده شوند. تخمینی که کتابخانه از انجام آن باز ماند.

علت تفاوت در دقت پیش بینی گوینده و رقم:

**تفاوت در تنوع کلاس ها:** تعداد و تنوع کلاس ها (speaker) ممکن است از تعداد و تنوع کلاس های اعداد متفاوت باشد. این تغییر ممکن است باعث شود که مدل با کارایی کمتری عمل کند، به خصوص اگر تعداد نمونه ها برای هر کلاس یکسان نباشد.

**تفاوت در ویژگی ها:** ممکن است ویژگی هایی که برای تشخیص اعداد مورد استفاده قرار گرفته بود، برای تشخیص صحبت کنندگان مناسب نباشد. برای تشخیص صحبت کنندگان، ممکن است نیاز به ویژگی های دیگر یا ترکیب های مختلفی از ویژگی ها باشد که مدل ما آنها را در نظر نگرفته است.

**مقیاس داده ها:** ممکن است داده ها مقیاس های مختلفی داشته باشند که می تواند به دقت ارزیابی مدل و معیار های دقت، بازیابی و امتیاز F1 تاثیر بگذارد. در برخی مواقع، نیاز به استاندارد سازی داده ها برای حل این مشکل و ارزیابی صحیح مدل وجود دارد.

تعداد hidden states از ۱۰ به ۵ نیز تغییر پیدا کرده است.