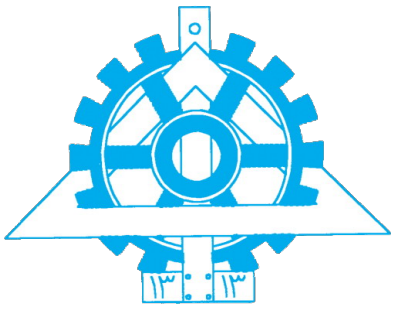


دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده علوم مهندسی
گروه هوش مصنوعی

گزارش پروژه نهایی یادگیری ماشین

پروژه نهایی برای دریافت درجهٔ در رشتهٔ مهندسی کامپیوتر
گرایش هوش مصنوعی

سید محمد عرفان موسوی منزه، غزل زمانی نژاد، نیما کمبرانی، سید مهدی موسوی

اساتید راهنما

دکتر اعرابی و دکتر ابوالقاسمی و دکتر توسلی پور

پاییز و زمستان ۱۴۰۲

فهرست مطالب

فهرست تصاویر

فصل ۱

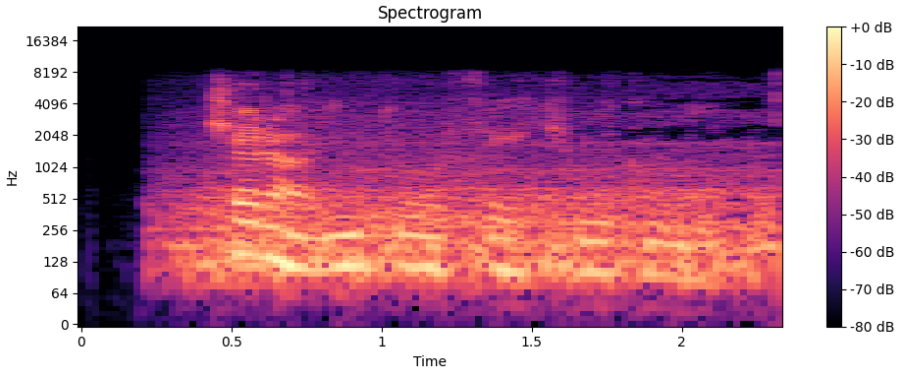
گزارش کار

۱.۱ پیش پردازش

در اولین قسمت از پروژه به تمیز کردن داده های صوتی در دست و همچنین استخراج ویژگی از آن می پردازیم.این کار را در چند قسمت انجام می دهیم.

۱.۱.۱ کوتاه کردن داده

در این مرحله، تلاش می‌کنیم با استفاده از توابع موجود، بخش‌هایی از سیگنال ورودی که دارای شدت و انرژی پایین‌تر از یک حد مشخص هستند را حذف کنیم. این کار باعث می‌شود که سکوت‌های میان کلمات یا در ابتدای صداها حذف شوند. بنابراین، سیگنال باقی‌مانده فقط شامل لحظاتی خواهد بود که در آن‌ها یک کلمه با معنی گفته شده است.



شکل ۱.۱: اسپکتوگرام سیگنال ورودی یکی از نمونه ها

۲.۱.۱ تبدیل فوریه کوتاه زمانی

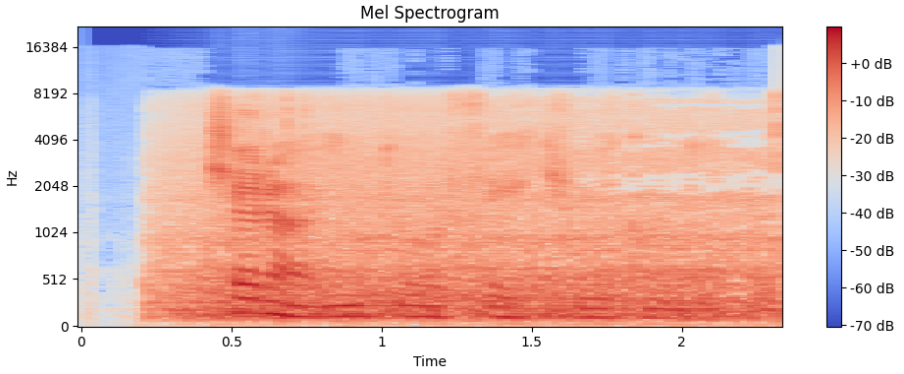
در ادامه، به استخراج تبدیل فوریه کوتاه زمانی [۱] می‌پردازیم. این تبدیل، همانطور که از نامش پیداست، سیگنال را در بازه‌های کوتاه زمانی به تبدیل فوریه می‌رساند. دو پارامتری که به این تبدیل داده می‌شود، طول پنجره‌ای است که تبدیل فوریه روی آن انجام می‌شود و همچنین مقدار حرکت این پنجره در هر گام. در نهایت، قدر مطلق خروجی این تبدیل را می‌گیریم تا به اندازه این تبدیل برسیم.

دلیل استفاده از تبدیل فوریه این است که این تبدیل به ما نمایشی از سیگنال در فضای فرکانسی می‌دهد. به عبارت دیگر، این تبدیل و خود سیگنال دوگان هم هستند و با استفاده از فضای فرکانسی، احتمالاً به جداپذیری بهتری خواهیم رسید. همچنین، مفاهیمی که در نمایش سیگنال صوت در زمان نمود ندارند، مثل بم بودن صدا، را می‌توان در این فضا نمایش داد. در این فضا، به راحتی می‌توانیم فیلتر طراحی کنیم یا از فیلترهای تعریف شده استفاده کنیم.

۳.۱.۱ فیلتر مل

در ادامه کار، از فیلتر مل [۲] استفاده می‌کنیم. این فیلتر بر اساس خصوصیات ادراکی انسان طراحی شده است و به خروجی فرکانسی مرحله قبلی اعمال می‌شود. در خروجی این مرحله، به اسپکتروگرام سیگنال می‌رسیم که توصیف سیگنال در طول زمان را نشان می‌دهد. حالا، تنها نیاز است که این خروجی را به مقیاس لگاریتمی ببریم

[1] STFT
[2] MEL

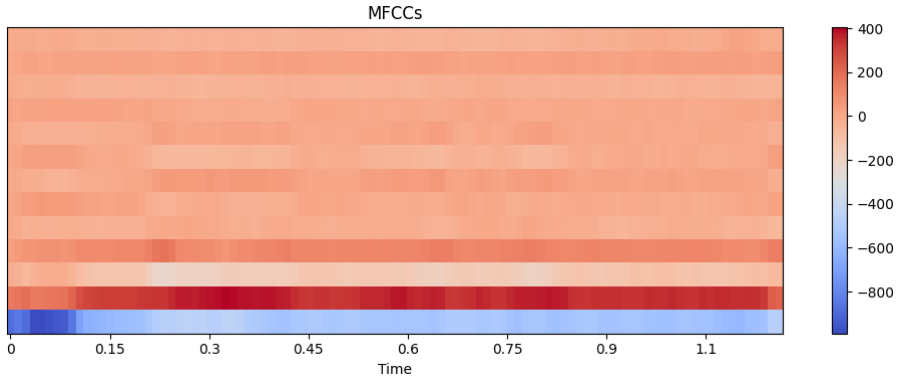


شکل ۲.۱: اسپکتوگرام بعد از اعمال فیلتر مل

تا بتوانیم شنوایی انسان را بهتر مدل کنیم.

فیلتر مل، که نامش از مخفف Mel Frequency Cepstral Coefficients گرفته شده، یکی از متداول‌ترین تکنیک‌ها در پردازش سیگنال‌های صوتی است. این فیلتر بر اساس این فرض که انسان صداهای نزدیک به هم را به خوبی تشخیص نمی‌دهد، طراحی شده است. بنابراین، فیلتر مل با تقسیم بندی فرکانس‌ها به صورت لگاریتمی، سعی در تقلید از شنوایی انسان دارد.

با استفاده از فیلتر مل، می‌توانیم ویژگی‌های مهم و کلیدی سیگنال صوتی را استخراج کنیم. این ویژگی‌ها می‌توانند در کاربردهای مختلفی مانند تشخیص گفتار، تشخیص لهجه، تشخیص هویت از طریق صدا و ... مورد استفاده قرار گیرند. در نهایت، با استفاده از این فیلتر، می‌توانیم سیگنال صوتی را به صورتی که بیشتر با شنوایی انسان همخوانی دارد، مدل کنیم.



شکل ۳.۱: اسپکتوگرام بعد از اعمال MFCC

۴.۱.۱ استخراج ویژگی

در مرحله نهایی، از سیگنال فیلتر شده استفاده می‌کنیم و با استفاده از توابع آماده، ویژگی MFCC را استخراج می‌کنیم. از این ویژگی به دلیل چندین دلیل استفاده می‌کنیم. اولاً، استفاده از این ویژگی باعث کاهش قابل توجه حجم داده‌ها می‌شود. دوماً، این ویژگی به تغییرات خارجی مثل نویز پس‌زمینه یا تغییرات جنسیت گوینده وابسته نیست. سوماً، این ویژگی به عنوان یکی از ویژگی‌های مهم در زمینه پردازش سیگنال‌های صوتی شناخته شده است .

MFCC یا همان Mel Frequency Cepstral Coefficients ، یک روش استخراج ویژگی برای تجزیه و تحلیل سخنرانی و صدا است. این روش سیگنال‌های صوتی خام را به یک نمایش فشرده تبدیل می‌کند که اطلاعات مهم فرکانسی و زمانی را ضبط می‌کند . این ویژگی‌ها می‌توانند در کاربردهای مختلفی مانند تشخیص گفتار، تشخیص لهجه، تشخیص هویت از طریق صدا و ... مورد استفاده قرار گیرند . در نهایت، با استفاده از این فیلتر، می‌توانیم سیگنال صوتی را به صورتی که بیشتر با شنوایی انسان همخوانی دارد، مدل کنیم .

۲.۱ طبقه بندی

برای طبقه بندی داده ها بر اساس جنسیت از دو سری طبقه بند مختلف استفاده کردیم. سری اول شامل طبقه بند های معرفی شده در کلاس است. شامل طبقه بند بیزی با فرض توزیع گاوسی و فرض ساده لوحی، طبقه بند بردار پشتیبان، درخت تصمیم و طبقه بند آدابوست است. در دسته دوم طبقه بند هایی را داریم که کمتر در کلاس معرفی شده اند. مانند، رندوم فارست، گردینت بوستینگ و پرسپترون چند لایه. قبل از انجام دسته بندی، از بردار ویژگی بر روی بعد دوم میانگین گرفتیم تا تمام داده ها به اندازه ثابت ۱۳ عدد ویژگی داشته باشند. به عنوان پیش پردازش بیشتر داده ها از PCA و Normalize کردن نیز استفاده کردیم منتها بدلیل کاهش عملکرد مدل بعد از این پیش پردازش ها، از آن ها در مدل نهایی استفاده نکردیم. از آنجا که توضیح داده ها به نفع داده های مزتبط با جنسیت مرد است، آموزش مدل با همین داده ها مدل را به سمت پیش بینی بیشتر داده ها به عنوان مرد می برد. برای جلوگیری از این مشکل از تکنیک آور سمپلینگ برای افزایش تعداد داده های مرتبط با کلاس زن استفاده کردیم. تکنیک SMOTE یا Synthetic Minority Oversampling Technique یک روش برای مقابله با مشکل نامتوازن بودن داده‌ها در مسائل طبقه‌بندی است. در بسیاری از مسائل واقعی، تعداد نمونه‌های یک کلاس

(کلاس اقلیت) بسیار کمتر از کلاس دیگر (کلاس اکثریت) است. این موضوع می‌تواند باعث شود الگوریتم‌های یادگیری ماشین به نامناسب عمل کنند و عملکرد ضعیفی روی کلاس اقلیت داشته باشند. روش SMOTE برای حل این مشکل، نمونه‌های سنتتیک یا مصنوعی برای کلاس اقلیت ایجاد می‌کند. این نمونه‌های سنتیک با استفاده از بین‌المللی بر روی نمونه‌های موجود کلاس اقلیت تولید می‌شوند.

۱.۲.۱ طبقه بند های معرفی شده در کلاس

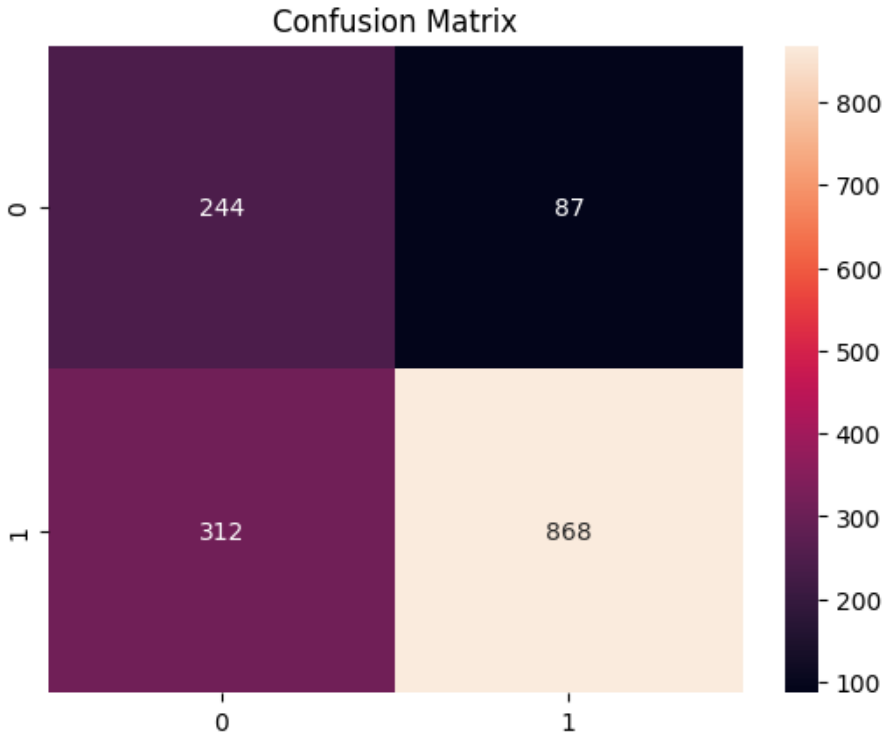
اولین طبقه بند معرفی شده در کلاس، مدل گاوسی مبتنی بر فرض ساده لوحانه بود. این مدل با وجود سادگی توانایی خوبی در دسته بندی داده ها دارد.

۱.۱.۲.۱ طبقه بند گاوسی

طبقه‌بند گاوسی یک الگوریتم یادگیری ماشین برای طبقه‌بندی است. این الگوریتم بر پایه توزیع احتمال گاوسی است و می‌تواند به عنوان پایه برای الگوریتم‌های یادگیری ماشین غیرپارامتریک پیچیده برای طبقه‌بندی و رگرسیون استفاده شود . در زیر نتایج استفاده از طبقه بند بیزی آورده شده است.

جدول ۱.۱: ارزیابی مدل بیزی

	Precision	Recall	F1-score	Support
۰	۴۴.۰	۷۴.۰	۵۵.۰	۳۳۱
۱	۹۱.۰	۷۴.۰	۸۱.۰	۱۱۸۰
Accuracy			۷۴.۰	۱۵۱۱
Macro avg	۶۷.۰	۷۴.۰	۶۸.۰	۱۵۱۱
Weighted avg	۸۱.۰	۷۴.۰	۷۶.۰	۱۵۱۱

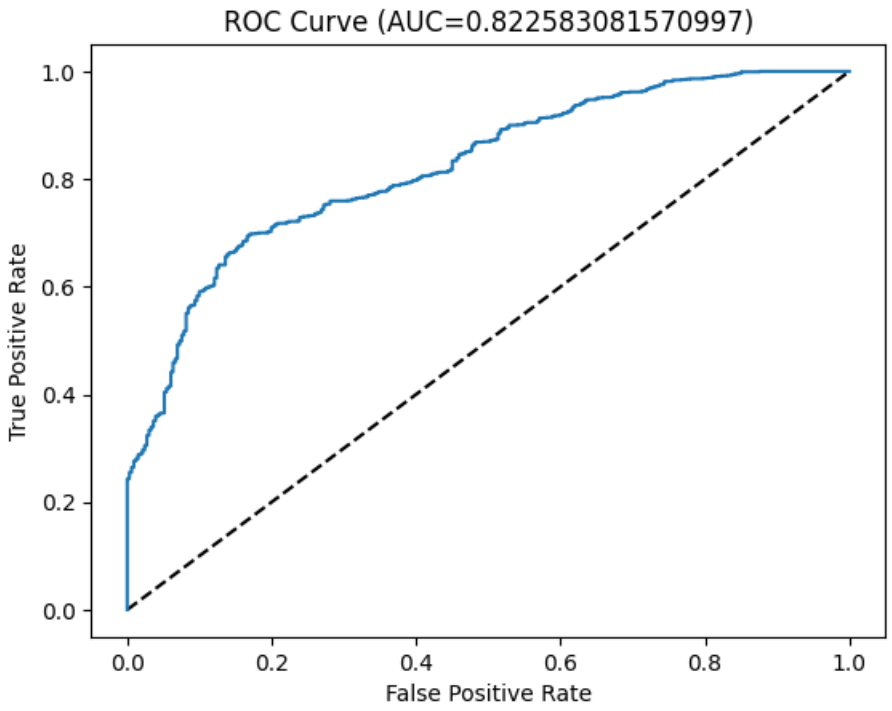


شکل ۴.۱: ماتریس درهم ریختگی طبقه بند بیزی

۲.۱.۲.۱ طبقه بند بردار پشتیبان

طبقه‌بند بردار پشتیبان SVM یک مدل یادگیری ماشینی است که برای حل مسائل طبقه‌بندی دو گروهی استفاده می‌شود. پس از ارائه مجموعه‌هایی از داده‌های آموزشی برچسب‌دار به مدل SVM برای هر دسته، آن‌ها قادر به دسته‌بندی متن جدید هستند.

SVM دو مزیت اصلی نسبت به الگوریتم‌های جدیدتر مانند شبکه‌های عصبی دارد: سرعت بیشتر و عملکرد بهتر با تعداد محدودی از نمونه‌ها (در هزارها)، که این موضوع باعث می‌شود الگوریتم برای مسائل طبقه‌بندی متن بسیار مناسب باشد. SVM با تلاش برای یافتن هایپرپلین که بیشترین فاصله را از نزدیک‌ترین نقاط کلاس‌های مختلف دارد، عمل می‌کند. اگر چنین هایپرپلینی وجود داشته باشد، به آن هایپرپلین حداکثر حاشیه یا حاشیه سخت گفته می‌شود. SVM همچنین قادر است تا با داده‌های غیرخطی کار کند. در این موارد، از ترفندهایی مانند "ترفند هسته" استفاده می‌شود تا داده‌ها را به فضایی با بعد بالاتر تبدیل کند که در آن می‌توان یک حاشیه خطی را یافت. برای این پروژه ما از هسته RBF برای تبدیل داده ها به فضای جدید استفاده کردیم.



شکل ۵.۱: منحنی مشخصه عملکرد سیستم برای طبقه بند بیزی

۳.۱.۲.۱ درخت تصمیم

درخت تصمیم یک الگوریتم یادگیری ماشین برای حل مسائل طبقه‌بندی و رگرسیون است. این الگوریتم یک ساختار درختی مانند جریان نما می‌سازد که هر گره داخلی آن یک ویژگی را نشان می‌دهد، هر شاخه قوانین را نشان می‌دهد و هر گره برگ نتیجه الگوریتم را نشان می‌دهد.

در طول آموزش، الگوریتم درخت تصمیم بهترین ویژگی را برای تقسیم داده‌ها بر اساس یک معیار مانند انتروپی یا ناخالصی گینی انتخاب می‌کند، که سطح ناخالصی یا تصادفی در زیرمجموعه‌ها را اندازه‌گیری می‌کند. هدف یافتن ویژگی‌ای است که بیشترین افزایش اطلاعات یا کاهش ناخالصی را پس از تقسیم ایجاد کند.

۴.۱.۲.۱ طبقه بند آدابوست

طبقه‌بند AdaBoost یک الگوریتم یادگیری ماشین است که برای حل مسائل طبقه‌بندی و رگرسیون استفاده می‌شود. AdaBoost یکی از اولین الگوریتم‌های تقویتی است که معرفی شده است. این الگوریتم با ترکیب

جدول ۲.۱: ارزیابی طبقه بند ماشین بردار پشتیبان

Support	F1-score	Recall	Precision	
۳۳۱	۶۲.۰	۸۴.۰	۴۹.۰	۰
۱۱۸۰	۸۴.۰	۷۵.۰	۹۴.۰	۱
۱۵۱۱	۷۷.۰			**Accuracy**
۱۵۱۱	۷۳.۰	۸۰.۰	۷۲.۰	**Macro avg**
۱۵۱۱	۷۹.۰	۷۷.۰	۸۴.۰	**Weighted avg**

جدول ۳.۱: ارزیابی طبقه بند درخت تصمیم

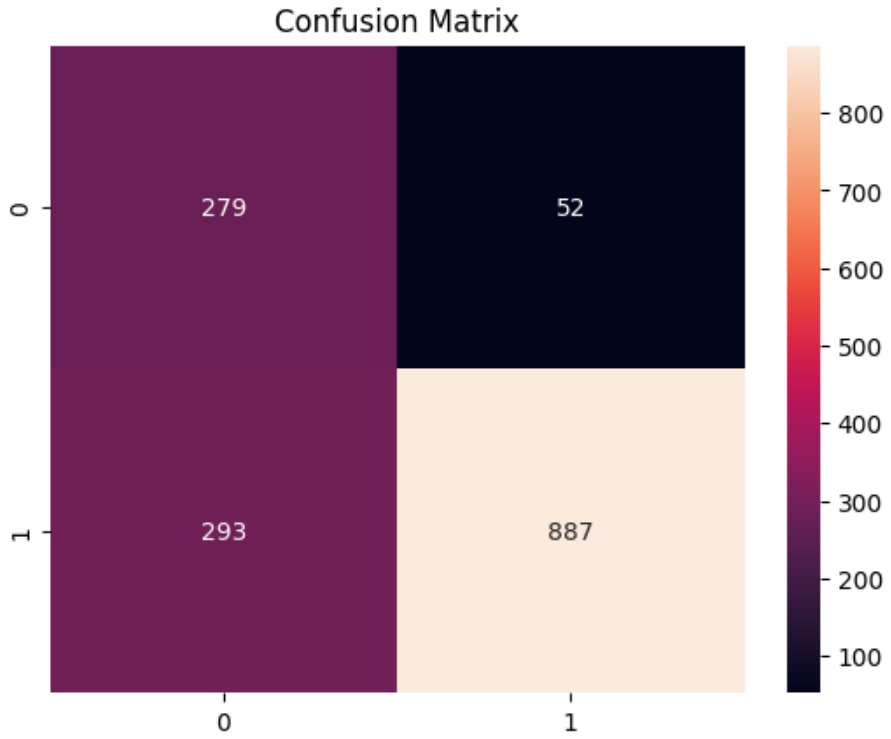
Support	F1-score	Recall	Precision	
۳۳۱	۸۸.۰	۹۰.۰	۸۵.۰	۰
۱۱۸۰	۹۶.۰	۹۶.۰	۹۷.۰	۱
۱۵۱۱	۹۴.۰			**Accuracy**
۱۵۱۱	۹۲.۰	۹۳.۰	۹۱.۰	**Macro avg**
۱۵۱۱	۹۴.۰	۹۴.۰	۹۵.۰	**Weighted avg**

چندین "طبقه‌بند ضعیف" یک "طبقه‌بند قوی" ایجاد می‌کند.

AdaBoost با شروع آموزش یک طبقه‌بند روی مجموعه داده اصلی و سپس آموزش نسخه‌های اضافی از طبقه‌بند روی همان مجموعه داده کار می‌کند، اما وزن نمونه‌هایی که به طور نادرست طبقه‌بندی شده‌اند تنظیم می‌شود تا طبقه‌بندهای بعدی بیشتر روی موارد سخت تمرکز کنند.

۲.۲.۱ سایر طبقه بند ها

علاوه بر طبقه بند های قبلی که در کلاس معرفی شد، سه طبقه بند دیگر نیز بررسی شده اند که در کلاس زیاد به آنها پرداخته نشده است.



شکل ۱.۶: ماتریس درهم ریختگی طبقه بند بردار پشتیبان

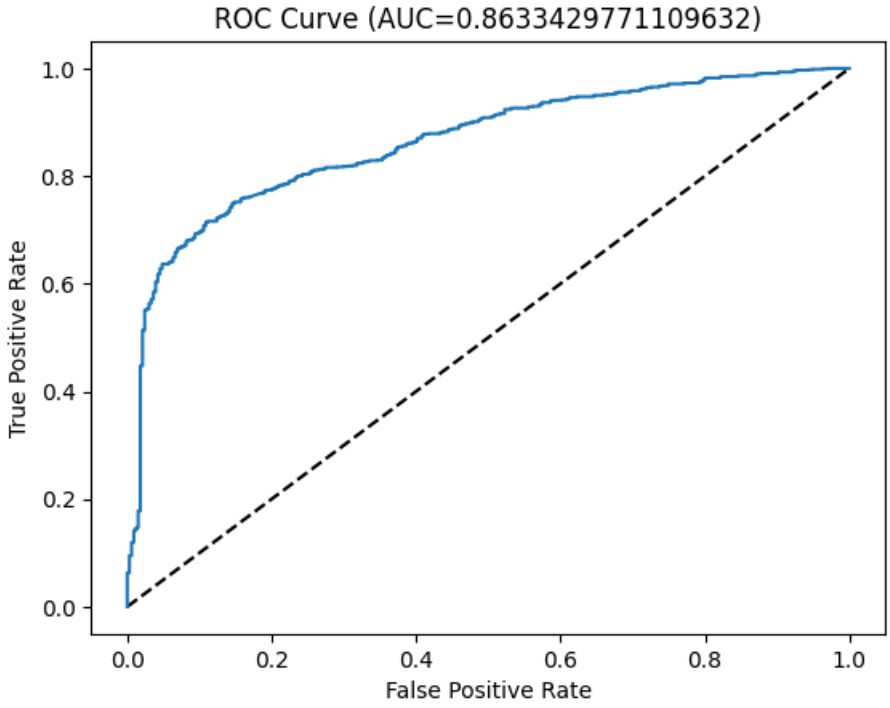
۱.۲.۲.۱ طبقه بند رندوم فارست

Random Forest یک الگوریتم یادگیری ماشین است که برای حل مسائل طبقه‌بندی و رگرسیون استفاده می‌شود. این الگوریتم ترکیبی از چندین درخت تصمیم است که با هم ترکیب شده‌اند تا یک "جنگل" ایجاد کنند.

در طول فرآیند آموزش، Random Forest چندین درخت تصمیم را بر اساس مجموعه داده‌های آموزشی می‌سازد. سپس، پیش‌بینی‌های هر درخت ترکیب می‌شوند تا یک پیش‌بینی نهایی ایجاد کنند. این فرآیند باعث می‌شود که Random Forest عملکرد بهتری نسبت به یک درخت تصمیم منفرد داشته باشد.

Random Forest همچنین از روش Bagging یا Bootstrap Aggregation استفاده می‌کند. در این روش، چندین نمونه آموزشی به طور تصادفی از مجموعه داده‌های آموزشی انتخاب می‌شوند (با جایگذاری مجدد) و سپس برای آموزش هر درخت استفاده می‌شوند.

Random Forest برای حل مسائل مختلفی از جمله طبقه‌بندی، رگرسیون، و کاهش ابعاد داده‌ها استفاده می‌شود. این الگوریتم به خاطر قابلیت‌هایی مانند تعدیل بیش‌برازش ، کار با داده‌های بزرگ، و قابلیت کار با



شکل ۷.۱: منحنی مشخصه عملکرد سیستم برای طبقه بند بردار پشتیبان

داده‌هایی که دارای ویژگی‌های نامرتبط یا گم‌شده هستند، محبوب است.

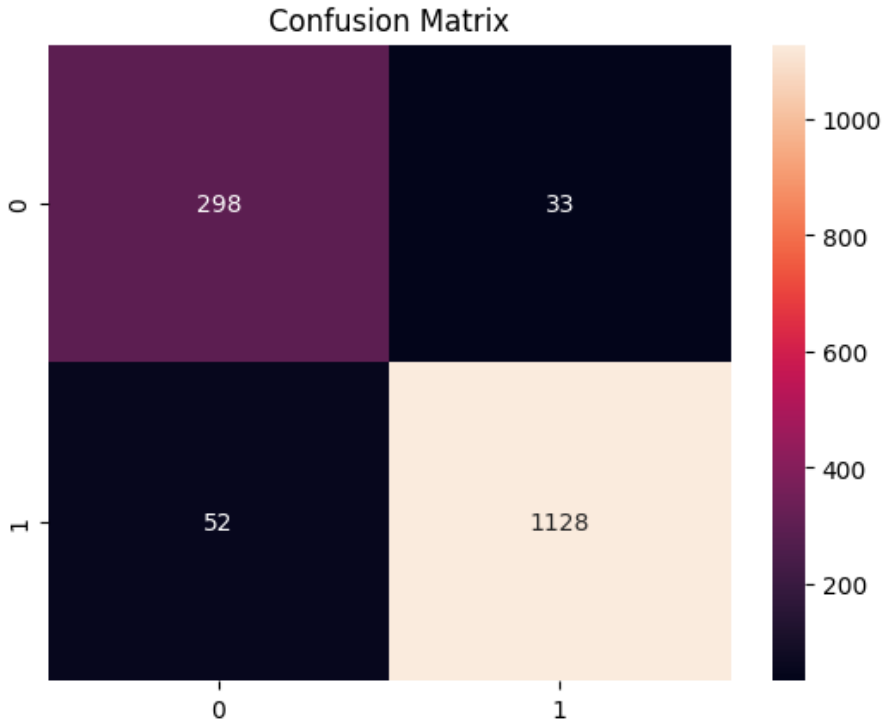
۲.۲.۲.۱ طبقه بند بوست گرادیانی

گرادیان بوست یک الگوریتم یادگیری ماشین است که برای حل مسائل طبقه‌بندی و رگرسیون استفاده می‌شود. این الگوریتم با ترکیب چندین "طبقه‌بند ضعیف" (معمولاً درختان تصمیم) یک "طبقه‌بند قوی" ایجاد می‌کند.

گرادیان بوست با ساختن مدل‌های پیش‌بینی ساده به صورت توالی کار می‌کند که هر مدل سعی می‌کند خطای باقی مانده از مدل قبلی را پیش‌بینی کند. در اینجا، یک مدل یادگیری ضعیف، مدلی است که کمی بهتر از پیش‌بینی تصادفی عمل می‌کند.

گرادیان بوست برای مسائل مختلف یادگیری ماشین، از جمله رگرسیون و طبقه‌بندی، بسیار مناسب است. این الگوریتم با ترکیب طبقه‌بندهای ضعیف در یک مجموعه، مانند درختان تصمیم، یک پیش‌بین قوی را ایجاد می‌کند.

گرادیان بوست با بهینه‌سازی وزن‌های مدل بر اساس خطاهای تکرارهای قبلی، به تدریج خطای پیش‌بینی را



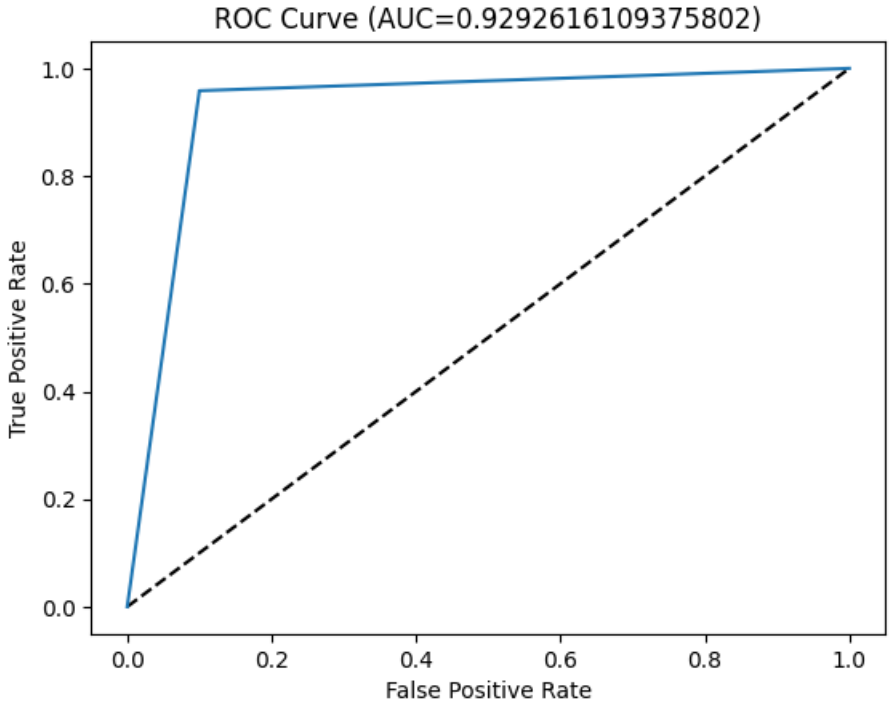
شکل ۸.۱: ماتریس درهم ریختگی طبقه بند درخت تصمیم

کاهش می‌دهد و دقت مدل را افزایش می‌دهد.

۳.۲.۲.۱ طبقه بند پرسپترون چندلایه

پرسپترون چند لایه یک نوع شبکه عصبی مصنوعی است که از یک سری نورون‌های کاملا متصل با تابع فعال‌سازی غیرخطی تشکیل شده است. این شبکه‌ها حداقل سه لایه دارند و قادر به تشخیص داده‌هایی هستند که به صورت خطی جدا نمی‌شوند.

MLP از یک لایه ورودی و یک لایه خروجی تشکیل شده است، اما ممکن است چندین لایه پنهان در بین این دو لایه وجود داشته باشد.



شکل ۹.۱: منحنی مشخصه عملکرد سیستم برای طبقه بند درخت تصمیم

۳.۲.۱ نتیجه گیری در طبقه بندی

در میان روش هایی غیر از شبکه عصبی، روش های مبتنی بر Ensemble Learning بهترین عملکرد را در مجموعه داده فعلی ما نشان داده اند. درخت تصمیم و شبکه عصبی پرسپترون عملکرد مشابهی داشته اند که این نشان دهنده قدرت طبقه بندی الگوریتم درخت تصمیم است، زیرا با وجود معماری بسیار ساده تر توانسته عملکردی مشابه با پرسپترون ارایه دهد. ماشین بردار پشتیبان بر خلاف انتظار عملکرد قوی نشان نداد. ما از هسته های خطی، RBF و چند جمله ای استفاده کردیم و بهترین عملکرد که مرتبط با RBF بود هم نتوانست جوابی نزدیک به مدل های دیگر بگیر. و در نهایت مدل بیزی ساده لوح پایین ترین دقت را ارایه داد که نشان می دهد ویژگی های استخراج شده با فرض ساده لوحانه سازگاری نداشته اند و باید مدل های بیزی دقیق تری تست شود.

جدول ۴.۱: ارزیابی طبقه بند آدابوست

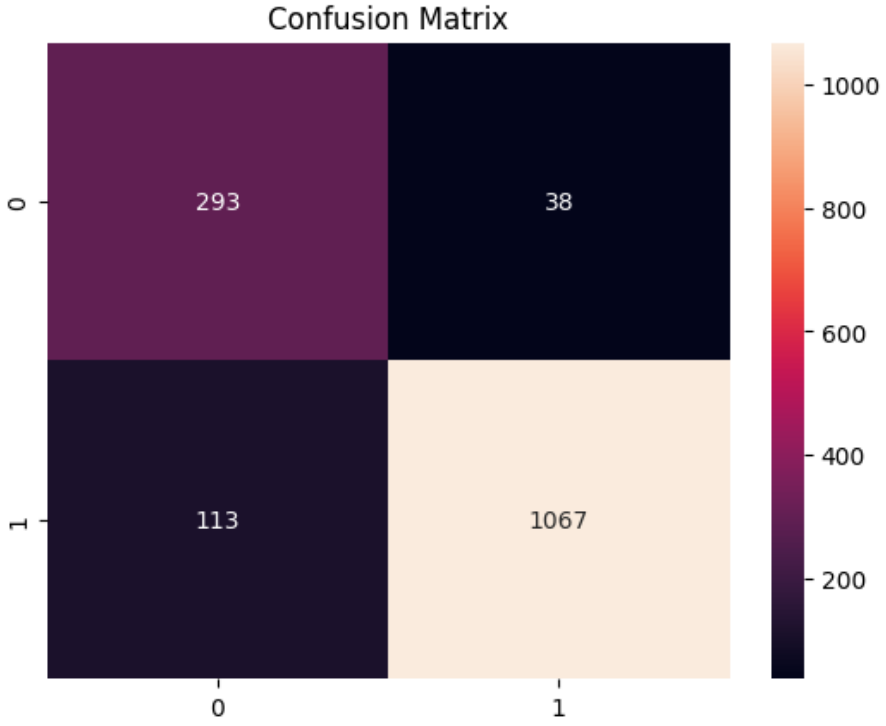
Support	F1-score	Recall	Precision	
۳۳۱	۸۰.۰	۸۹.۰	۷۲.۰	۰
۱۱۸۰	۹۳.۰	۹۰.۰	۹۷.۰	۱
۱۵۱۱	۹۰.۰			**Accuracy**
۱۵۱۱	۸۶.۰	۸۹.۰	۸۴.۰	**Macro avg**
۱۵۱۱	۹۰.۰	۹۰.۰	۹۱.۰	**Weighted avg**

جدول ۵.۱: ارزیابی طبقه بند رندوم فارست

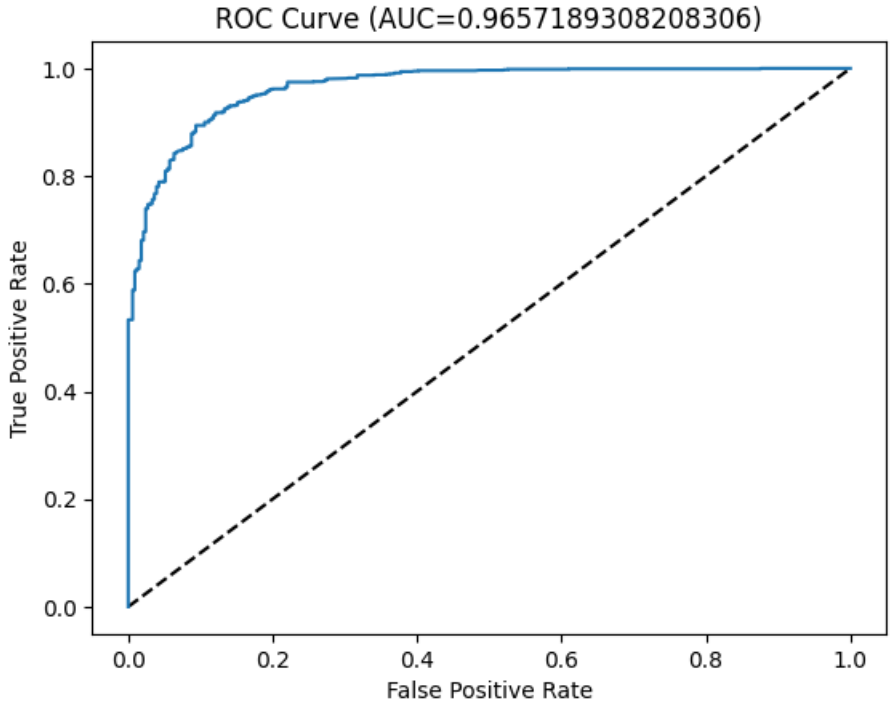
Support	F1-score	Recall	Precision	
۳۳۱	۹۶.۰	۹۵.۰	۹۷.۰	۰
۱۱۸۰	۹۹.۰	۹۹.۰	۹۹.۰	۱
۱۵۱۱	۹۸.۰			**Accuracy**
۱۵۱۱	۹۷.۰	۹۷.۰	۹۸.۰	**Macro avg**
۱۵۱۱	۹۸.۰	۹۸.۰	۹۸.۰	**Weighted avg**

جدول ۶.۱: ارزیابی طبقه بند گرادیان بوستینگ

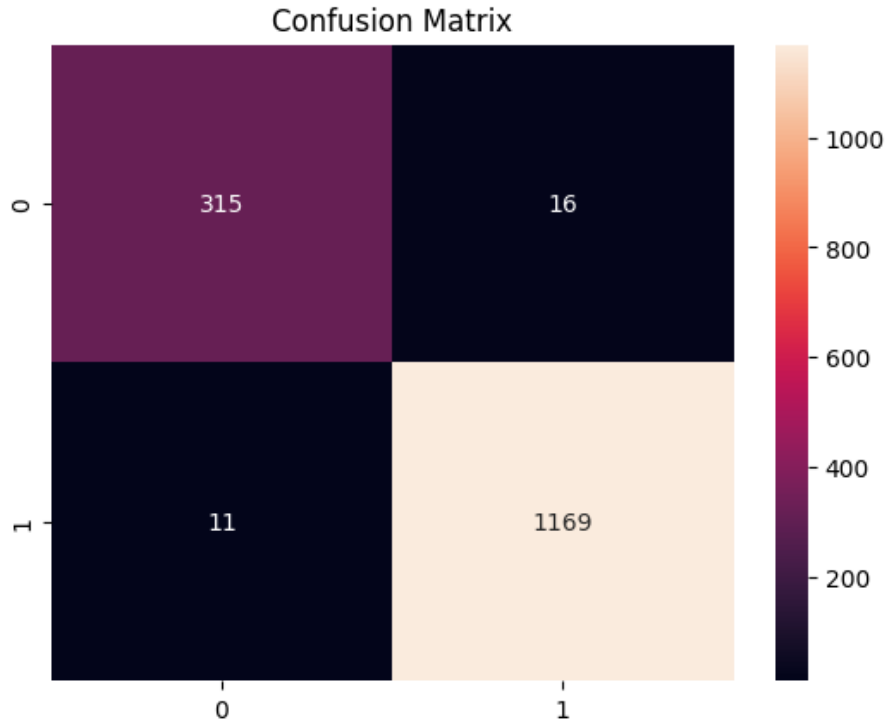
Support	F1-score	Recall	Precision	
۳۳۱	۸۸.۰	۹۳.۰	۸۴.۰	۰
۱۱۸۰	۹۷.۰	۹۵.۰	۹۸.۰	۱
۱۵۱۱	۹۵.۰			**Accuracy**
۱۵۱۱	۹۲.۰	۹۴.۰	۹۱.۰	**Macro avg**
۱۵۱۱	۹۵.۰	۹۵.۰	۹۵.۰	**Weighted avg**



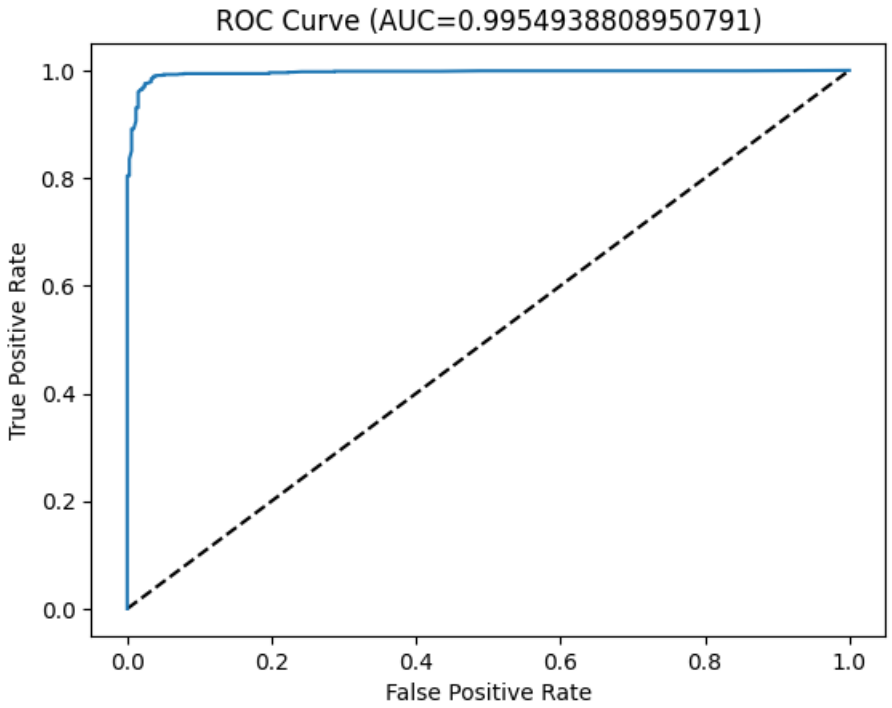
شکل ۱۰.۱: ماتریس درهم ریختگی طبقه بند آدابوست



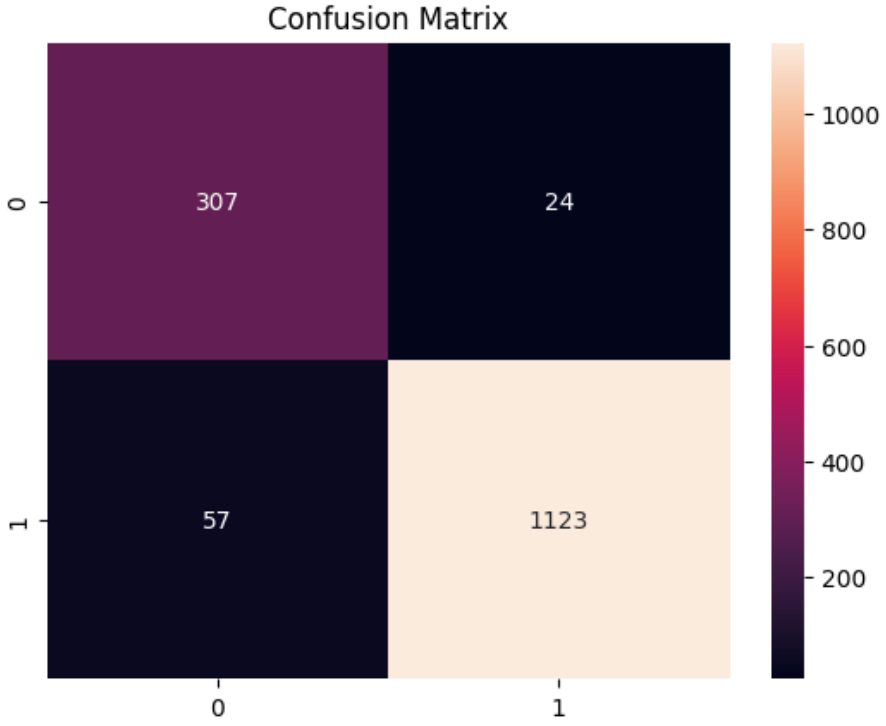
شکل ۱۱.۱: منحنی مشخصه عملکرد سیستم برای طبقه بند آدابوست



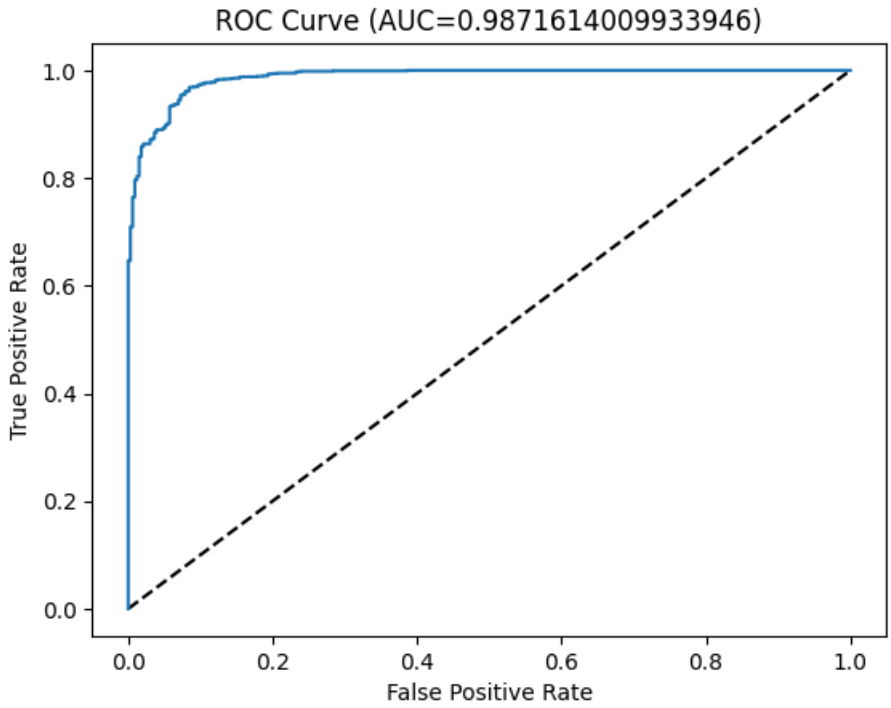
شکل ۱۲.۱: ماتریس درهم ریختگی طبقه بند رندوم فارست



شکل ۱۳.۱: منحنی مشخصه عملکرد سیستم برای طبقه بند رندوم فارست



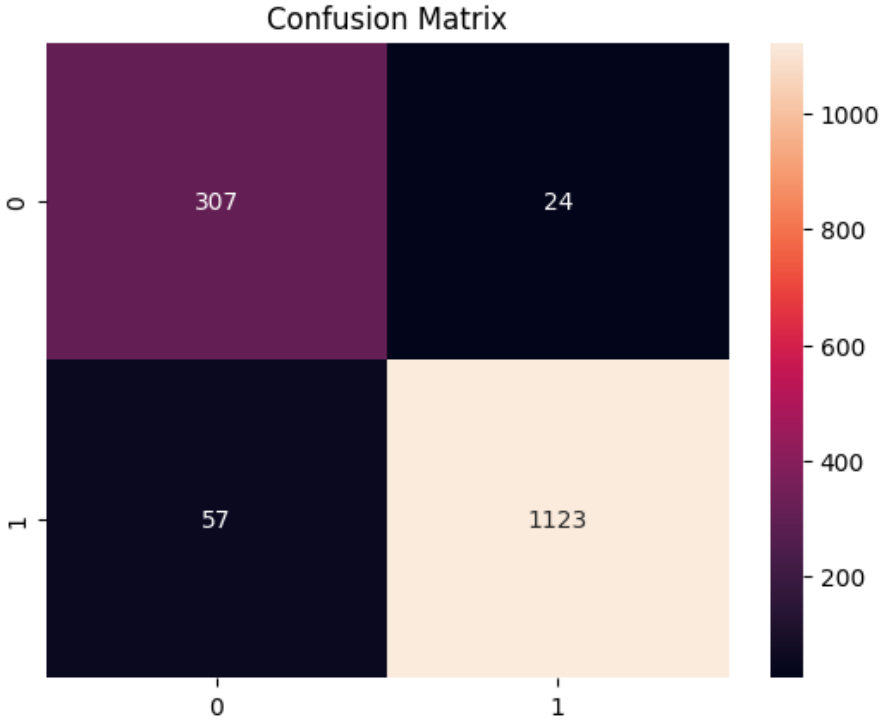
شکل ۱۴.۱: ماتریس درهم ریختگی طبقه بند گرادیان بوست



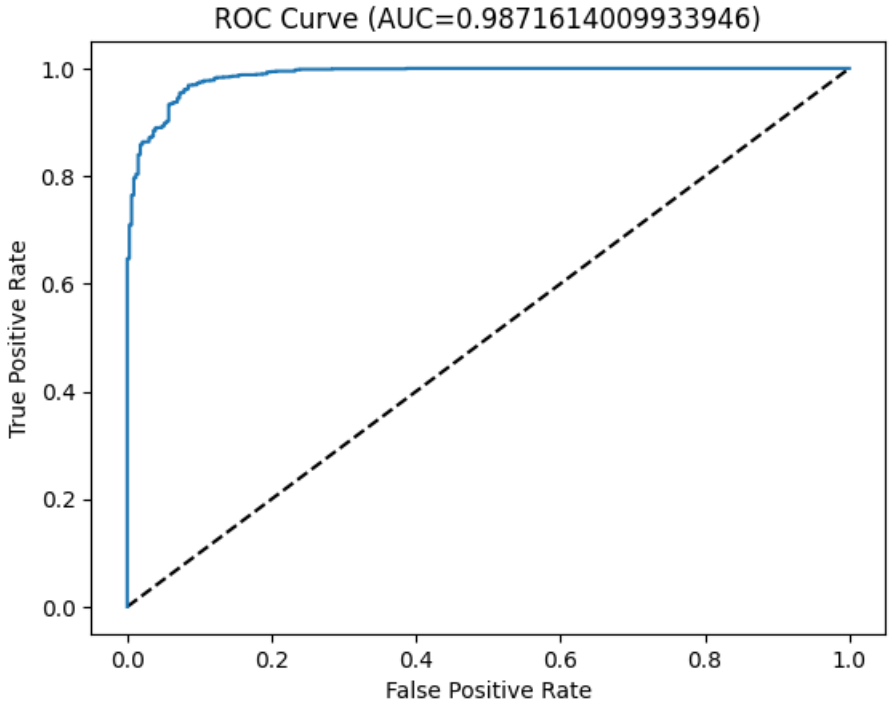
شکل ۱۵.۱: منحنی مشخصه عملکرد سیستم برای طبقه بند گرادیان بوست

جدول ۷.۱: ارزیابی طبقه بند پرسپترون چند لایه

Support	F1-score	Recall	Precision	
۳۳۱	۹۱.۰	۸۷.۰	۹۶.۰	۰
۱۱۸۰	۹۸.۰	۹۹.۰	۹۶.۰	۱
۱۵۱۱	۹۶.۰			**Accuracy**
۱۵۱۱	۹۴.۰	۹۳.۰	۹۶.۰	**Macro avg**
۱۵۱۱	۹۶.۰	۹۶.۰	۹۶.۰	**Weighted avg**



شکل ۱۶.۱: ماتریس درهم ریختگی طبقه بند پرسپترون چندلایه



شکل ۱۷.۱: منحنی مشخصه عملکرد سیستم برای طبقه بند پرسپترون چند لایه

۳.۱ خوشه بندی

خوشه‌بندی به فرایند دسته‌بندی نمونه‌های مشابه در یک دسته بدون استفاده برچسب‌های متناظر گفته می‌شود. در فرایند دسته‌بندی نمونه‌ها با توجه به تفاوت طول فایل‌های صوتی مختلف و همچنین وجود نویز در داده‌های ورودی ابتدا نیاز است تا داده‌های ورودی تا حد ممکن تمیز شده و سپس از آن‌ها استخراج ویژگی صورت گیرد. برای این مسئله در بخش پیشین با استفاده از تابع mfcc در کتابخانه librosa ویژگی‌های مختلفی با توجه به صوت و تبدیل فوریه صوت استخراج شده است که در بخش‌های دیگر مورد استفاده قرار می گیرد. سپس ویژگی‌های بدست آمده با استفاده از روش‌های کاهش بعد فشرده سازی می شوند. در نهایت الگوریتم‌های خوشه‌بندی بر روی ویژگی‌های بدست آمده اعمال شده و نتایج آن نمایش داده می‌شود.

۱.۳.۱ پردازش ویژگی

با توجه به ساختار داده‌های صوتی، برای استخراج ویژگی در هر مرحله پنجره کوچکی از صوت انتخاب شده و برای آن بخش ویژگی‌های متناظر محاسبه می‌شوند. ویژگی استخراج شده دارای ۱۳ بعد برای هر پنجره است. در نتیجه با توجه به طول اولیه صوت، تعداد پنجره‌ها و متعاقبا طول ویژگی‌ها متفاوت خواهد بود. با توجه به اینکه برای اعمال الگوریتم‌های خوشه‌بندی نیاز است که طول ویژگی‌ها برای همه نمونه‌ها برابر باشد، باید استفاده از روشی برای همه نمونه‌ها ویژگی با طول ثابت محاسبه شود. برای محاسبه ویژگی واحد در اینجا ما از عملگر میانگین استفاده می‌کنیم. در نتیجه تمام مقادیر مختلف برای صوت با یکدیگر میانگین گرفته شده و به ازای هر صوت ۱۳ ویژگی خواهیم داشت.

۲.۳.۱ نرمال سازی ویژگی

در مرحله بعد با توجه به حساسیت الگوریتم‌های خوشه‌بندی به فاصله و مقیاس، نیاز داریم که این ویژگی‌ها نرمال‌سازی شوند. به عنوان مثال، در مورد ویژگی‌های صوتی، شدت یک ویژگی ممکن است مقیاسی بسیار بزرگتر از دیگری داشته باشد. دیگر اینکه بعضی از الگوریتم‌های روش‌های خوشه‌بندی زمانی که ویژگی‌ها به مقیاس مشابهی باشند، سریع‌تر همگرا می‌شوند. این موضوع به خصوص برای الگوریتم‌هایی مانند K-Means مهم است که فاصله بین نقاط داده عامل مهمی است. همچنین اعمال نرمال‌سازی، تفسیر نتایج خوشه‌بندی را

آسان‌تر می‌کنند. بدون نرمال‌سازی، ویژگی‌های با مقیاس بزرگتر ممکن است نتایج خوشه‌بندی را تسلط کنند و اهمیت نسبی هر ویژگی در فرآیند خوشه‌بندی را مشخص کنند. در نرمال سازی تلاش می‌شود تا برای هر ویژگی با استاندارد سازی میانگین ۰ و واریانس ۱ داشته باشیم.

۳.۳.۱ فشرده سازی ویژگی ها و نمایش توزیع داده‌ها

گاهی اوقات یک مجموعه‌داده ممکن است دچار Curse of Dimensionality شود که در آن تعداد ویژگی‌ها بسیار بیشتر از تعداد مشاهدات است. اساساً در فضاهای با ابعاد بالا با مشاهدات کم، تفکیک مشاهدات در ابعاد فوق دشوار می‌شود. همچنین، با توجه به اینکه داده‌ها به صورت دو بعدی نمایش داده می شوند نیاز است که ترتیب اهمیت ویژگی‌ها محاسبه و دو ویژگی پر اهمیت تر جهت نمایش توزیع داده‌ها بدست آورده شود. بدین دلایل، جهت کاهش ابعاد دو الگوریتم PCA و t-SNE مورد استفاده قرار گرفته است. در این حالت تعداد ابعاد با استفاده از دو الگوریتم گفته شده از ۱۳ به ۳ کاهش پیدا کرده و سپس دو بعد اول برای نمایش در نظر گرفته شده است.

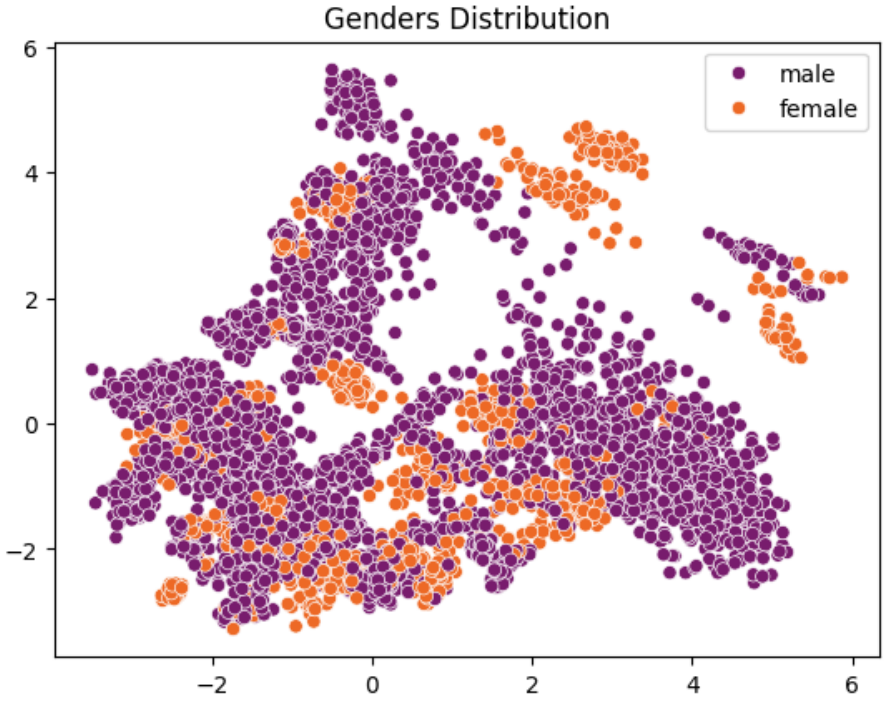
PCA ۱.۳.۳.۱

این روش کاهش بعد با یافتن یک تبدیل خطی بین ویژگی‌های ورودی با استفاده از بردار ویژه و مقدار ویژه سعی می کند تا نمونه‌های ورودی را بر روی بردارهای با بیشترین واریانس نگاشت کند. این کار می تواند به حذف ویژگی‌های با تاثیر کمتر در جدایی‌پذیری و همچنین حذف نویز کمک کند. در ادامه، نمایش نمونه‌ها در فضای دوبعدی برای دسته‌های مختلف برچسب‌ها قرار داده شده است. در این حالات تعداد ابعاد با استفاده از روش PCA کاهش بعد داده شده اند.

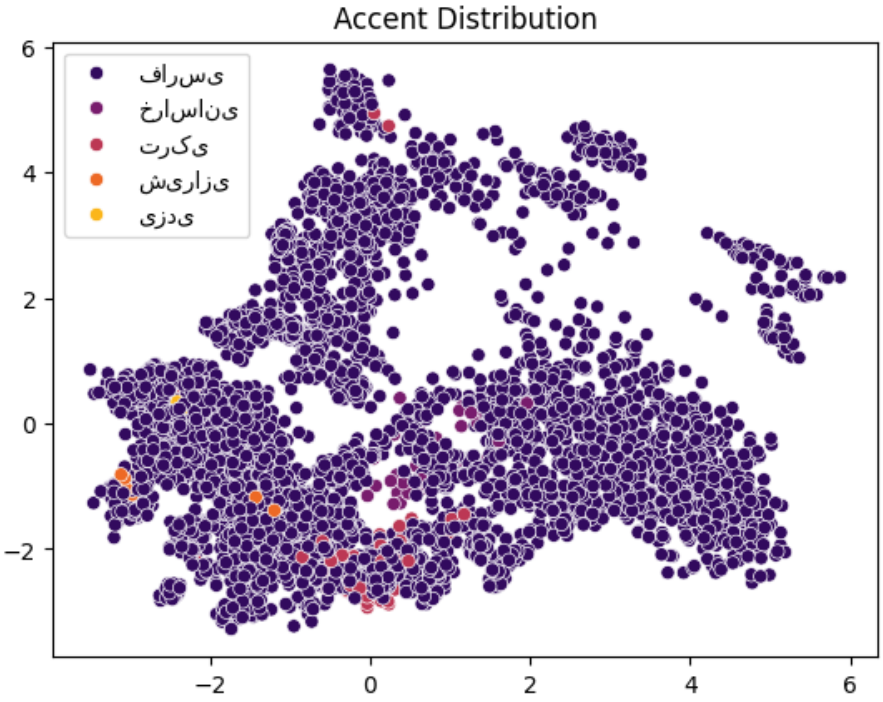
T-SNE ۴.۳.۱

این روش با یادگیری پارامترهایی سعی می‌کند تا یک تبدیل غیر خطی جهت کاهش بعد ارائه کند که ساختار فاصله نمونه‌ها را در ابعاد پایین تر نسبت حالت اولیه حفظ کند. با توجه به غیر خطی بودن این تبدیل برای یادگیری روابط پیچیده‌تر در داده‌ها مناسب است. این تبدیل باعث با توجه پارامتر تعداد همسایگی مورد توجه جهت حفظ

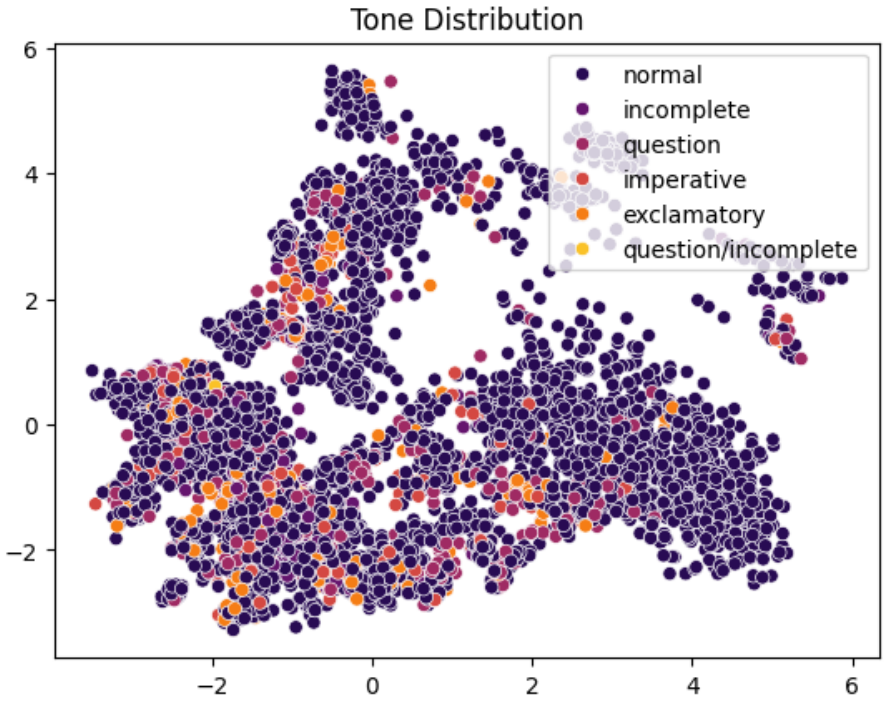
رابطه فاصله می‌تواند باعث ایجاد نقاط جدا از یکدیگر یا دسته‌های متعدد از نمونه‌های نزدیک به یکدیگر شود. در ادامه توزیع داده‌ها پس از کاهش ابعاد با استفاده از این روش قرار داده شده است.



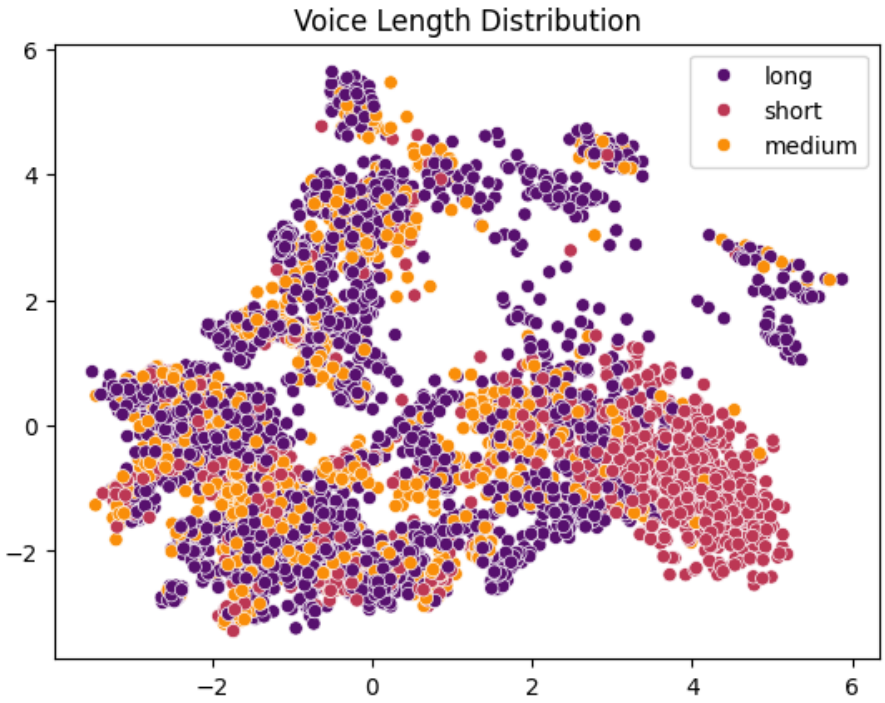
شکل ۱۸.۱: توزیع نمونه‌ها بر اساس جنسیت گوینده، پس از اعمال PCA . دسته‌های یک جنسیت در کنار هم قرار گرفته اند که ممکن است بخاطر مشابه بودن گوینده باشد.



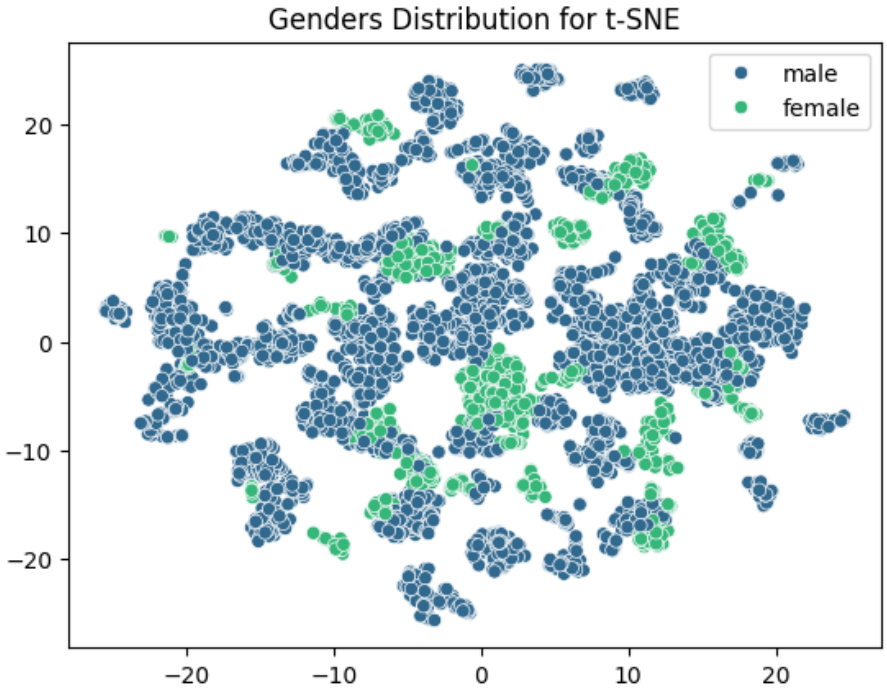
شکل ۱۹.۱: توزیع نمونه‌ها بر اساس لهجه‌های مختلف، پس از اعمال PCA . در اکثر نمونه‌ها لهجه فارسی است.



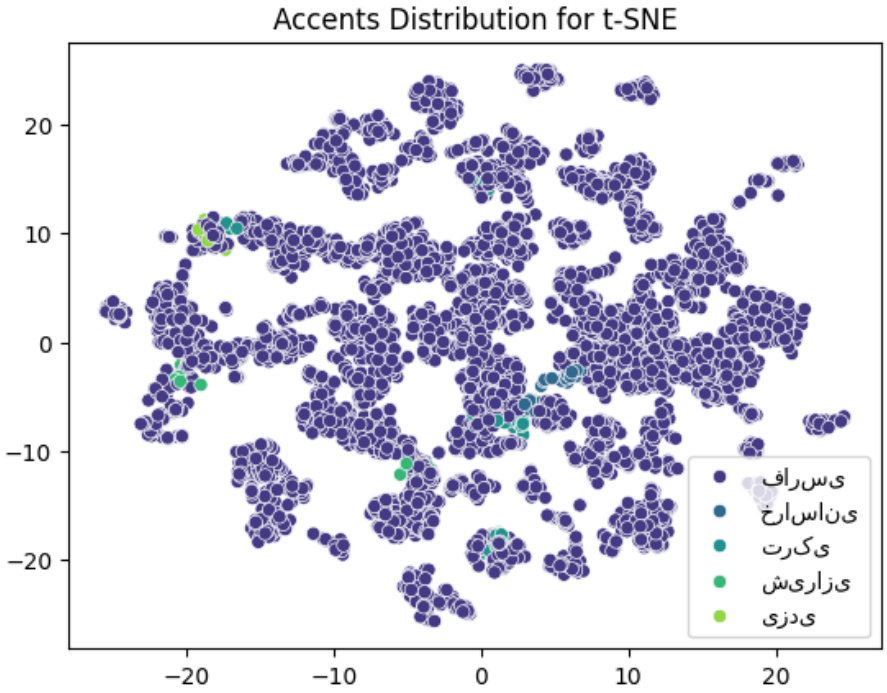
شکل ۲۰.۱: توزیع انواع جملات خوانده شده با توجه به تن هر یک، پس از اعمال PCA



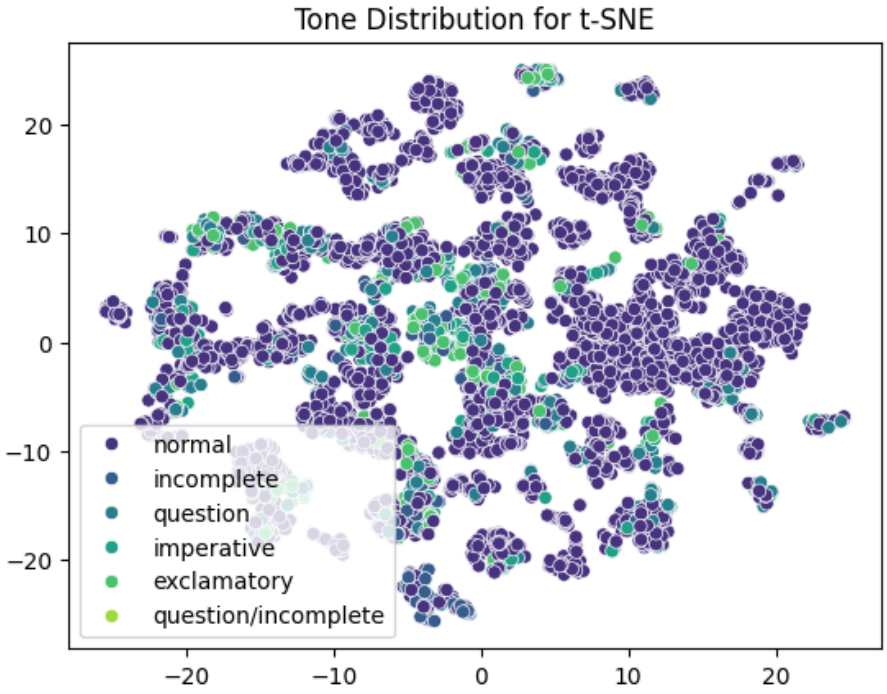
شکل ۲۱.۱: توزیع نمونه‌ها بر اساس طول ویژگی‌ها، پس از اعمال PCA . نمونه‌های با طول کمتر از ۱۵۰ کوتاه، بین ۱۵۰ تا ۲۵۰ متوسط و بیشتر از ۲۵۰ بلند برچسب زده شده اند. اکثر نمونه‌های با طول کوتاه در کنار یکدیگر در گوشه راست تصویر قرار گرفته اند.



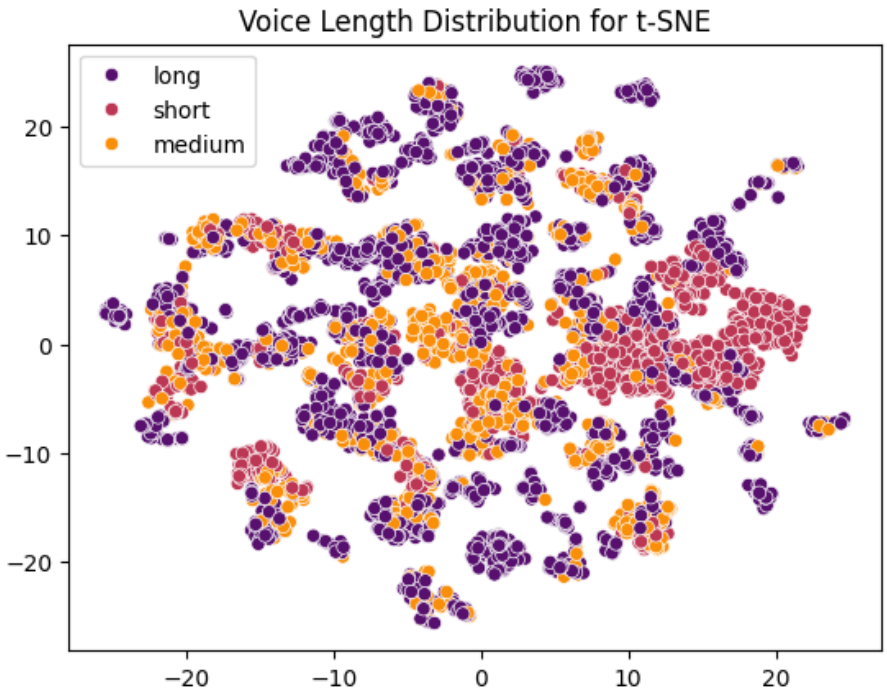
شکل ۲۲.۱: توزیع نمونه‌ها بر اساس جنسیت گوینده، پس از اعمال t-SNE



شکل ۲۳.۱: توزیع نمونه‌ها بر اساس لهجه‌های مختلف، پس از اعمال t-SNE



شکل ۲۴.۱: توزیع انواع جملات خوانده شده با توجه به تن هر یک، پس از اعمال t-SNE



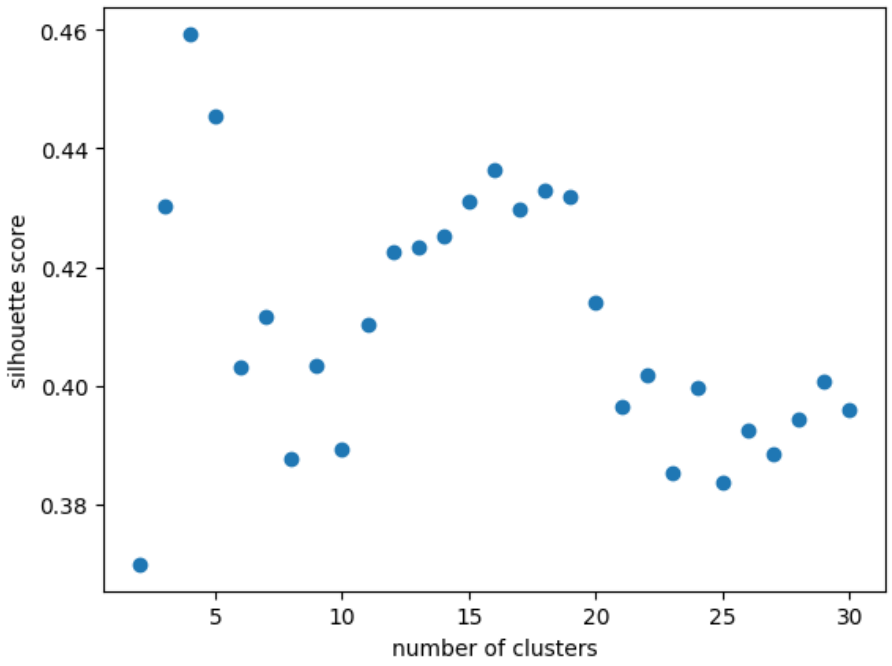
شکل ۲۵.۱: توزیع نمونه‌ها بر اساس طول ویژگی‌ها، پس از اعمال t-SNE . نمونه‌های با طول کمتر از ۱۵۰ کوتاه، بین ۱۵۰ تا ۲۵۰ متوسط و بیشتر از ۲۵۰ بلند برچسب زده شده اند. اکثر نمونه‌های با طول کوتاه در کنار یکدیگر در گوشه راست تصویر قرار گرفته اند.

۵.۳.۱ خوشه بندی

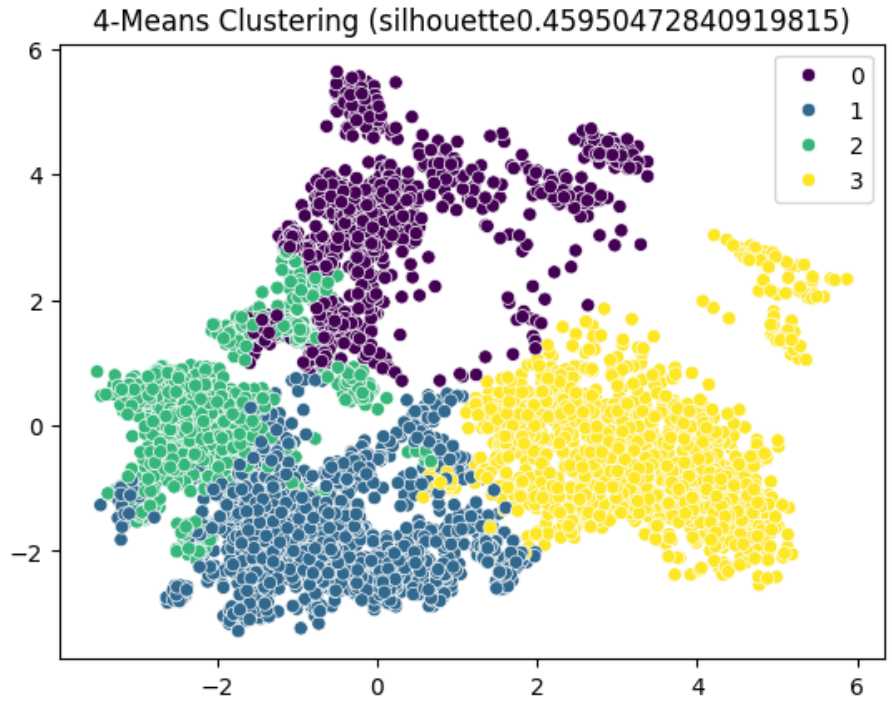
برای خوشه‌بندی داده‌ها ۲ روش k-means و Agglomerative Hierarchical مورد بررسی قرار گرفته اند. K-means یک روش ساده و از نظر محاسباتی بهینه است و می‌تواند برای مجموعه‌داده بزرگ به کار رود. خوشه‌های حاصل از این روش با مرکزشان نمایش داده می‌شوند و تفسیرپذیری راحتی نیز دارند. در این الگوریتم تنها پارامتر تعداد خوشه‌ها باید مشخص شود که آن هم از طریق امتیاز silhouette و یا روش elbow قابل محاسبه است. عیب k-means این است که خوشه‌ها را به صورت کروی در نظر می‌گیرد و ممکن است اگر توزیع داده‌ها به گونه‌ای باشد که خوشه‌ها شکل غیرمعمول داشته باشند، چندان عملکرد مناسبی نداشته باشد. Agglomerative Hierarchical که به عنوان روش دوم مورد استفاده قرار گرفته، یک سلسله مراتب از ترکیب خوشه‌ها را در قالب دندروگرام نمایش می‌دهد و بدین ترتیب دید خوبی از روند تجمیع به ما می‌دهد. در این روش نیازی به مشخص کردن تعداد خوشه‌ها از ابتدای الگوریتم نیست و می‌توانیم تعداد خوشه‌ها را در انتهای کار، با جدا کردن نمودار دندروگرام از یک ارتفاع خاص انتخاب کنیم. از مزایای دیگر می‌توان به منعطف بودن linkage ، یعنی روش محاسبه فاصله بین خوشه‌ها برای ترکیب، اشاره کرد. این الگوریتم نسبت به شکل خوشه‌ها فرضی ندارد و می‌تواند برای خوشه‌ها با شکل غیرمعمول مناسب باشد. ایراد این روش آن است که از نظر محاسباتی پیچیده است، مخصوصا برای مجموعه‌داده‌های بزرگ. اکنون به توضیح هر یک از روش‌ها می‌پردازیم.

K-Means ۱.۵.۳.۱

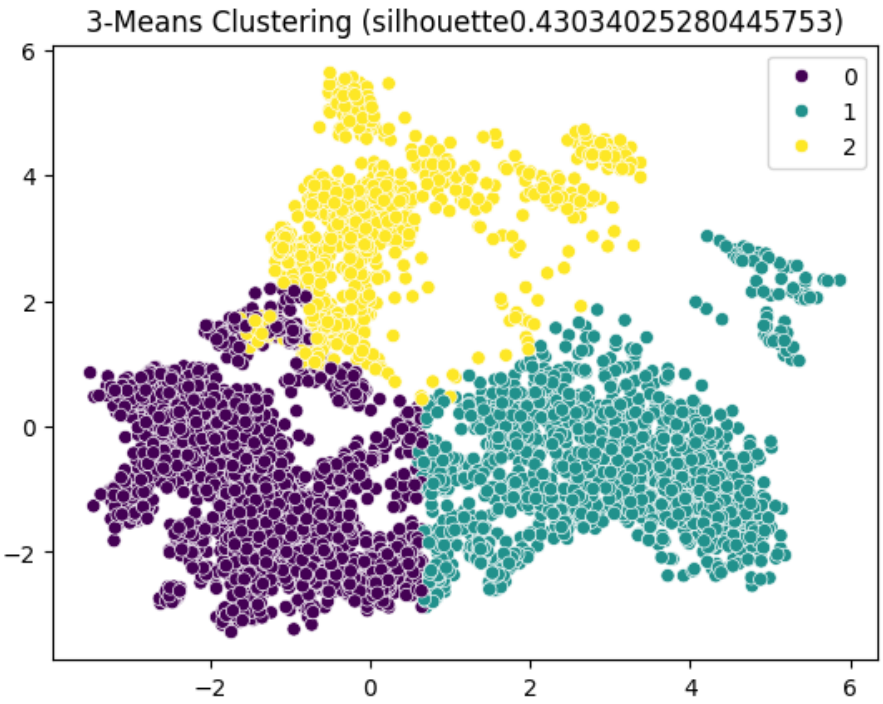
این روش با تعیین تعداد خوشه‌ها به اندازه k سعی می کند تا برای هر خوشه یک مرکز که از میانگین نمونه‌های داخل دسته محاسبه می‌شود قرار دهد. سپس هر نمونه به خوشه‌ای که به مرکز آن نزدیکتر است اختصاص داده می‌شود. برای محاسبه عملکرد الگوریتم خوشه‌بندی از معیار silhouette score استفاده می کنیم. در این حالت ابتدا الگوریتم خوشه‌بندی بر روی داده‌ها با تعداد خوشه‌ها در بازه ۲ تا ۳۰ اجرا شده و برای هر یک از اجراها معیار کیفیت محاسبه می‌شود. در ادامه روند تغییر مقدار silhouette و نمایش ۳ آزمایش مختلف قرار داده شده است. با توجه به اینکه برای عملیات خوشه‌بندی از ۳ ویژگی استفاده شده است اما در نمایش تنها دو بعد نمایش داده شده است خوشه‌ها کمی در نمایش تا حدی در هم تنیدگی دارند.



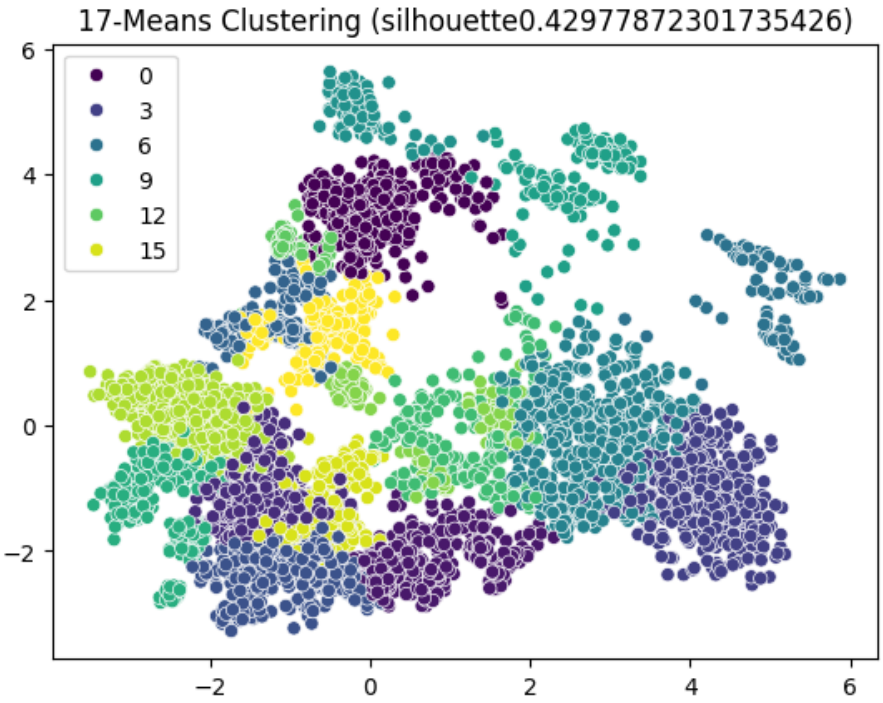
شکل ۲۶.۱: روند تغییر مقدار silhouette score با افزایش تعداد دسته‌ها در الگوریتم k-means



شکل ۲۷.۱: نمایش دسته‌های مختلف در الگوریتم k-means در حالت k=4 . این حالت بیشتری خروجی معیار را دارد.



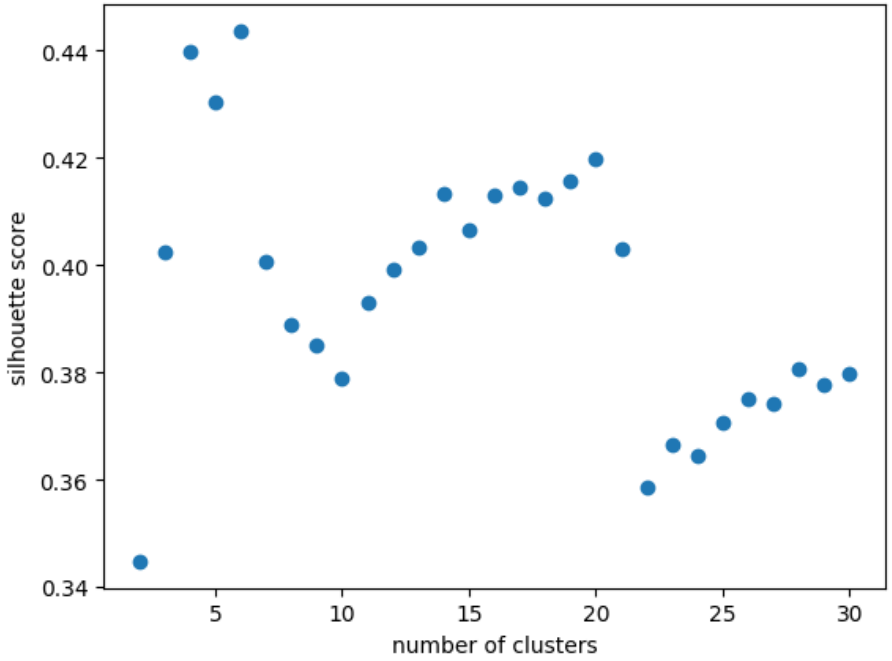
شکل ۲۸.۱: نمایش دسته‌های مختلف در الگوریتم k-means در حالت k=3 .



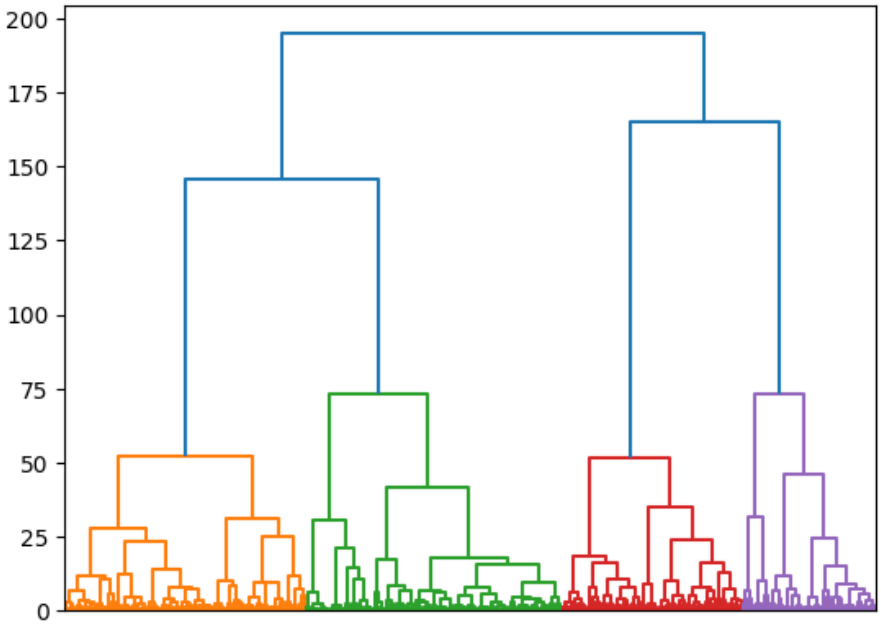
شکل ۲۹.۱: نمایش دسته‌های مختلف در الگوریتم k-means در حالت k=17 .

۲.۵.۳.۱ Hierarchical Agglomerative

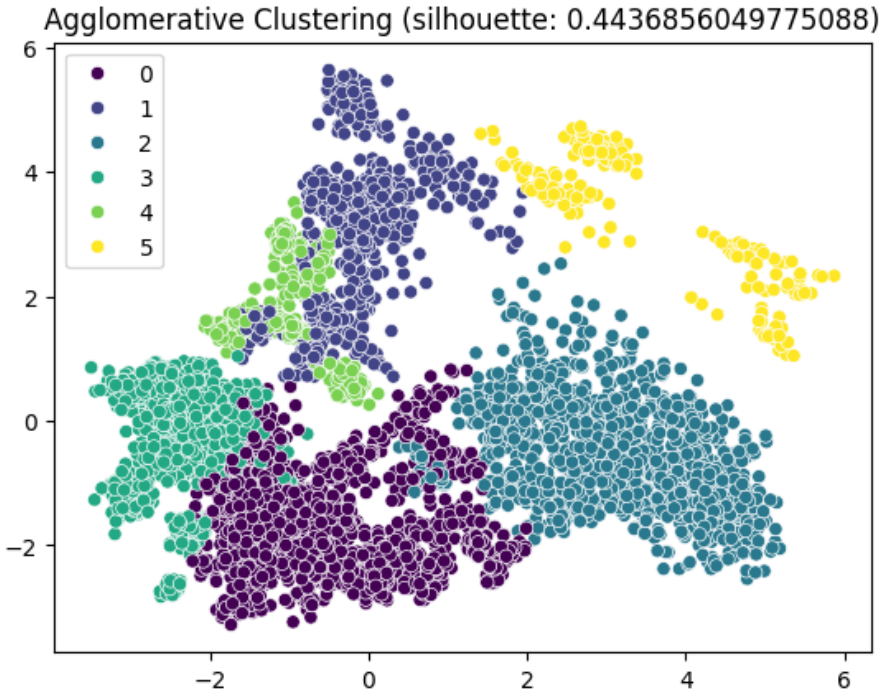
این الگوریتم خوشه‌بندی به صورت سلسله مراتبی نمونه‌ها با کمترین فاصله و بیشترین شباهت را با یک دیگر ترکیب می‌کند تا در نهایت پس از مراحل مختلف به یک دسته واحد برسد. با توجه به نیاز می‌توان تعداد دسته‌ها را در مقدار خاصی موقف کرد. برای نمایش سلسله مراتب انتخاب این در این روش از رسم دندروگرام استفاده کرده ایم. در این حالت ترتیب ادغام و شباهت دسته‌ها در هنگام ترکیب شدن مشخص می‌شود. در ادامه نمودار تغییر معیار silhouette و دسته‌بندی داده برای تعداد دسته‌های مختلف نمایش داده شده است.



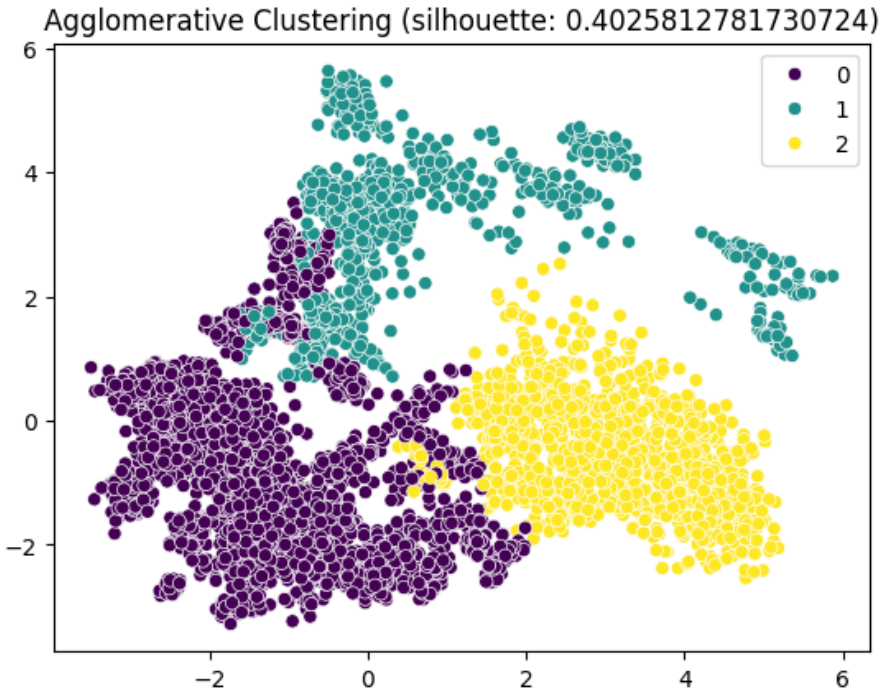
شکل ۳۰.۱: نمودار تغییرات معیار ارزیابی عملکرد خوشه‌بندی با افزایش تعداد دسته‌ها در الگوریتم سلسله مراتبی



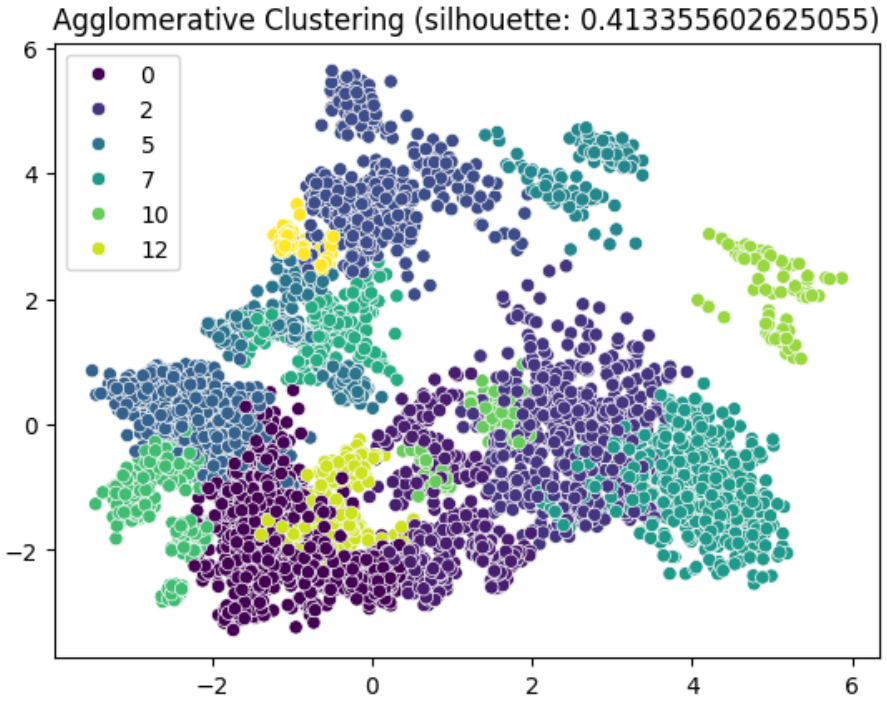
شکل ۳۱.۱: نمایش دندروگرام برای داده‌ها در کاهش بعد به ۳ با استفاده از روش PCA با توجه به نمودار دندروگرام و همچنین معیار Silhouette مقادیر ۴و۶ برای تعداد نمونه ها بهترین عملکرد را خواهند داشت. در این حالات فاصله بین دسته ها بیشترین حالت و هر دسته تا حد ممکن یکپارچه خواهد بود.



شکل ۳۲.۱: نمایش دسته‌های مختلف در الگوریتم سلسله مراتبی با تعداد دسته برابر با ۶. با توجه به وجود ۳ بعد برای خوشه بندی خوشه ی ۴ در نمایش تا دو تکه نشان داده شده است. دسته ۵ نیز با توجه به قرارگیری به صورت جدا از بقیه داده به عنوان یک خوشه در نظر گرفته شده است که بخش زیادی از گویندگان در این دسته خانوم هستند.



شکل ۳۳.۱: نمایش دسته‌های مختلف در الگوریتم سلسله مراتبی با تعداد دسته برابر با ۳. در این حالت با توجه به تعداد پایین دسته ها دو دسته آبی و بنفش به دارای همپوشانی بوده اند.



شکل ۳۴.۱: نمایش دسته‌های مختلف در الگوریتم سلسله مراتبی با تعداد دسته برابر با ۱۴. در این حالت تعداد دسته ها زیاد تر از حد نیاز شده است و تعداد اعضای هر دسته تا حدی نامتوازن شده است. هر چند همچنان الگوریتم توانسته تا حدی با توجه به توزیع جنسیت و دیگر کلاس ها مانند طول صوت خوشه بندی را انجام دهد.

فصل ۲

Automatic Speach Recognition(ASR)

۱.۲ مقدمه

تشخیص خودکار صحبت به فرایند تبدیل فایل صوتی از گفتار یک فرد و تبدیل آن به متن قابل‌خواندن و معادل با فایل صوتی اولیه گفته می‌شود. در اینجا تبدیل صوت به متن در سه مرحله پیش‌پردازش، آموزش و ارزیابی انجام شده است که در ادامه به توضیح هر کدام می‌پردازیم.

۲.۲ پیش‌پردازش

اولین مرحله از پیش‌پردازش یکسان‌سازی حروف موجود در متون مجموعه‌داده است. تعداد کاراکترهای موجود در مجموعه‌داده قبل از انجام هرگونه پردازشی، ۱۶۳ عدد بود. برای عملکرد بهتر مدل، لازم است کاراکترهای عربی را به فارسی تبدیل کنیم؛ به طور مثال داده «نتیجهٔ تطهیرکنندهٔ سفرهای زیارتی» به «نتیجه ی تطهیرکننده ی سفرهای زیارتی» تبدیل می‌شود. همچنین بعضی از کاراکترها از جمله علائم نگارشی و کاراکترهای اضافه را حذف می‌کنیم. با انجام این پیش‌پردازش‌ها، تعداد کاراکترهای منحصر بفرد به ۴۷ می‌رسد. الفبا را برای استفاده‌های بعدی در یک فایل ذخیره می‌کنیم. از توکنایزر و استخراج‌کننده ویژگی کتابخانه Wav2Vec استفاده می‌کنیم و داده‌ها را با اعمال processor آماده می‌کنیم. پس از حذف داده‌های بیشتر از ۱۵ ثانیه، داده پردازش‌شده

را در درایو ذخیره می‌کنیم.

۳.۲ آموزش

ابتدا داده‌های پردازش‌شده را از درایو لود می‌کنیم. از مدل از پیش‌آموخته Vec۲Wav استفاده می‌کنیم ولی قسمت feature_encoder آن را فریز می‌کنیم تا وزن‌های این بخش به روزرسانی نشود. حین آموزش، مقدار validation loss بی نهایت می‌شد که با اضافه کردن خط زیر این مشکل را برطرف کردیم:

model.config.ctc_zero_infinity = True

مدل را در ۲ ایپاک، با نرخ آموزش e-4، و با اندازه batch ۱ آموزش می‌دهیم. فرایند آموزش ۳ ساعت به طول انجامید. در ابتدا مقدار خطای آموزش ۱۷٫۰۷ و خطای ارزیابی ۱۱٫۹۰ بود. WER معیار دیگریست که برای ارزیابی مدل مورد استفاده قرار گرفت. این معیار، اختلاف بین دنباله مرجع (متن اصلی صوت) و دنباله خروجی (پیش‌بینی شده) را به ازای تعداد کلمات اندازه‌گیری می‌کند. فرمول آن به صورت زیر است:

$$WER = \frac{(S + D + I)}{N} \tag{۱.۲}$$

S تعداد خطاهای جایگزینی

D تعداد خطاهای حذف

I تعداد خطاهای اضافه

N تعداد کلمات کل در مرجع

معیار WER از گام رو به کاهش رفت و در نهایت به مقدار ۰٫۵۰۶ رسید. در جدول ۱.۲ به صورت خلاصه گزارش مقدار خطا و WER مشاهده می‌شود.

WER	Validation Loss	Training Loss	Step
1.000000	11.907510	17.074800	۲۰
1.000000	3.109851	5.678500	۴۰
...	...	...	...
0.999813	2.356913	2.608600	۴۲۰
0.991504	1.798767	2.243500	۴۴۰
...	...	...	...
0.521427	0.552796	0.696100	۹۲۰
0.506934	0.553852	0.747700	۹۴۰

جدول ۱.۲: معیارهای آموزش و ارزیابی

۴.۲ ارزیابی

در قسمت آخر به صورت تصادفی ۲۰ داده از داده‌های آزمون را انتخاب می‌کنیم و پیش‌بینی مدل را بدست می‌آوریم. در جدول ۲.۲، ۵ نمونه از آنها را مشاهده می‌کنیم. به نظر می‌رسد مدل می‌تواند تا حد خوبی خروجی را پیش‌بینی کند اما در برخی نمونه‌ها خروجی مدل با مرجع متفاوت است. دلایل احتمالی عبارتند از:

- ابتدا یا انتهای صوت به طور کامل ضبط نشده و به همین دلیل مدل کلمه را ناقص پیش‌بینی کرده (نمونه ۱).
- در جمع‌آوری مجموعه‌داده اشتباهاتی رخ داده و مرجع برخی از سطرهای داده با صوت منطبق نیست (نمونه ۲).
- لهجه فرد منجر به سخت‌تر شدن کار مدل شده و نتوانسته بعضی حروف را به درستی تخمین بزند (نمونه ۳، فرد با لهجه ترکی)
- در زبان فارسی تعداد زیادی از حروف آوای یکسان دارند ولی با کاراکترهای متفاوت نوشته می‌شوند. بزرگترین دلیل پیش‌بینی اشتباه مدل، این حروف هستند (نمونه ۴ و ۵).

شماره	پیش‌بینی	مرجع
۱	ن میتوانم با کداماتی که به تور تصادفی انتخاب می‌کنم	من میتوانم با کلماتی که به طور تصادفی انتخاب می‌کنم
۲	هر چند بازار کارق برق در ایران رو به رشته است اما همچنان چالش هایی نیز در این هوزه وجود دارد	افرادی که بتوانند این چالشها را مدیریت کنند در بازار کار برق در ایران موفقیت‌آمیز خواهند بود
۳	امیشه محربان خواهم ماند حتتا اگر کسی قدر مهربهنیم را نداند	همیشه مهربان خواهم ماند حتی اگر کسی قدر مهربانیم را نداند
۴	آدم قروب خورشید را دوست می‌دارد	ادم غروب خورشید را دوست می‌دارد
۵	من سدای قدم خاهش را می‌شنوم	من صدای قدم خواهش را می‌شنوم

جدول ۲.۲: پیش‌بینی و مرجع برای چند نمونه

در آخر، بر روی کل داده آزمون، پیش‌بینی مدل را بدست می‌آوریم. سپس مقادیر ویژگی‌های جنسیت، لهجه و تن صدا را یکدست می‌کنیم. اکنون همبستگی میان هر یک از ویژگی‌ها با مقدار WER را حساب می‌کنیم. به طور کلی برای محاسبه همبستگی با یک ویژگی، WER را برای هر یک از مقادیر آن ویژگی حساب می‌کنیم و سپس از معیار Point-Biserial Correlation استفاده می‌کنیم. این ضریب زمانی استفاده می‌شود که یکی از متغیرها پیوسته و دیگری categorical باشد. مقدار آن از ۱- تا ۱ متغیر است. مقدار ۱- نشان‌دهنده همبستگی منفی کامل، ۱ نشان‌دهنده همبستگی مثبت کامل و ۰ نشان‌دهنده عدم همبستگی است. مقدار همبستگی‌ها در جدول ۳.۲ قابل مشاهده است.

برای جنسیت، مرد را به ۰ و زن را به ۱ تبدیل کردیم. مقدار WER برای مرد 0.44 و برای زن 0.39 است. WER با جنسیت رابطه عکس دارد؛ بدین معنی که با افزایش مقدار جنسیت، میزان خطا کاهش یافته.

برای لهجه، فارسی را به ۰، شیرازی به ۱، ترکی به ۲، خراسانی به ۳ و یزدی به ۴ نگاشت کردیم. WER با لهجه رابطه مستقیم دارد؛ بدین معنی که با افزایش مقدار لهجه (دور شدن از لهجه فارسی)، میزان خطا افزایش یافته است. در مجموعه داده آزمون، ۱۱۲۴ عدد از دادگان لهجه فارسی دارند و در مجموع ۲۹ داده مربوط به ۴ لهجه دیگر هستند. چون مدل حین آموزش نیز از این دادگان بسیار کمتر از فارسی دیده، انتظار می‌رود در پیش‌بینی لهجه‌هایی غیر از فارسی خطای بیشتری داشته باشد و مقدار همبستگی نیز شاهدی بر همین ادعاست.

برای تن صدا، نرمال را به ۰، سوالی به ۱، تعجبی به ۲، دستوری به ۳ و ناقص به ۴ نگاشت کردیم. WER با تن رابطه عکس دارد؛ بدین معنی که با افزایش مقدار تن، میزان خطا کاهش یافته.

ویژگی	همبستگی با WER
جنسیت	-1
لهجه	0.8
تن	-0.44

جدول ۳.۲: همبستگی ویژگی‌ها با WER