



به نام خدا



دانشگاه تهران

دانشکده مهندسی برق و کامپیووتر

یادگیری ماشین

گزارش نهایی پروژه

-امیرحسین قاسمی- حدیثه مصباح- امیرپویا کارخانه یوسفی-

محمد مهدی سلمانی زارچی

نام و نام خانوادگی

۸۱۰۱۰۱۰۳۲-۸۱۰۱۰۲۲۵۳-۸۱۰۱۰۰۴۴۰-۸۱۰۱۰۲۱۷۴

شماره دانشجویی

۱۴۰۲/۱۱/۱۰

تاریخ ارسال گزارش

# فهرست

۹	.....	مقدمه
۹	.....	۱-پیش‌پردازش داده‌ها
۱۴	.....	۲-طبقه‌بندی
۱۴	.....	۲-۱-مقدمه
۱۴	.....	۲-۲-بررسی آماری داده‌ها
۱۶	.....	۲-۳- تقسیم داده‌های آموزش و تست
۱۶	.....	۲-۴- روش‌های طبقه‌بندی
۱۷	.....	Linear SVM-۲-۴-۱
۱۷	.....	Naïve Bayes-۲-۴-۲
۱۸	.....	RBF SVM-۲-۴-۳
۱۹	.....	Logistic Regressions-۲-۴-۴
۲۰	.....	MLP-۲-۴-۵
۲۱	.....	Ensemble Learning-۲-۴-۶
۲۲	.....	۲-۵- نرمال‌سازی
۲۳	.....	۲-۶- کاهش ابعاد
۲۴	.....	PCA-۲-۶-۱
۲۴	.....	LDA-۲-۶-۲
۲۵	.....	۲-۷- بررسی نتایج اجرای طبقه‌بندی‌های معرفی شده
۲۵	.....	Linear SVM-۲-۷-۱
۲۶	.....	Naïve Bayes-۲-۷-۲
۲۷	.....	RBF SVM-۲-۷-۳
۲۸	.....	Logistic Regression-۲-۷-۴
۲۹	.....	MLP-۲-۷-۵
۳۰	.....	Ensemble Method-۲-۷-۶
۳۲	.....	Linear SVM-۲-۷-۷
۳۳	.....	Naïve Bayes-۲-۷-۸
۳۴	.....	RBF SVM-۲-۷-۹

۳۵	..... بدون کاهش بعد و بدون نرمال سازی Logistic Regression-۲-۷-۱۰
۳۶	..... MLP-۲-۷-۱۱ بدون کاهش بعد و بدون نرمال سازی
۳۷	..... Ensemble Method-۲-۷-۱۲ بدون کاهش بعد و بدون نرمال سازی
۳۸	..... ۲-۸-کاهش بعد با استفاده از PCA
۳۹	..... ۲-۸-۱ Linear SVM با کاهش ابعاد و با نرمال سازی
۴۰	..... ۲-۸-۲ Naïve Bayes با کاهش ابعاد و با نرمال سازی
۴۱	..... ۲-۸-۳ RBF SVM با کاهش ابعاد و با نرمال سازی
۴۲	..... ۲-۸-۴ Logistic Regression با کاهش ابعاد و با نرمال سازی
۴۳	..... ۲-۸-۵ MLP با کاهش ابعاد و با نرمال سازی
۴۴	..... ۲-۸-۶ Ensemble Method با کاهش ابعاد و با نرمال سازی
۴۵	..... ۲-۸-۷ Linear SVM با کاهش ابعاد و بدون نرمال سازی
۴۶	..... ۲-۸-۸ Naïve Bayes با کاهش ابعاد و بدون نرمال سازی
۴۷	..... ۲-۸-۹ RBF SVM با کاهش ابعاد و بدون نرمال سازی
۴۸	..... ۲-۸-۱۰ Logistic Regression با کاهش ابعاد و بدون نرمال سازی
۴۹	..... ۲-۸-۱۱ MLP با کاهش ابعاد و بدون نرمال سازی
۵۰	..... ۲-۸-۱۲ Ensemble Method با کاهش ابعاد و بدون نرمال سازی
۵۱	..... ۲-۹-کاهش بعد با استفاده از LDA
۵۱	..... ۲-۹-۱ Linear SVM با کاهش ابعاد و با نرمال سازی
۵۲	..... ۲-۹-۲ Naïve Bayes با کاهش ابعاد و با نرمال سازی
۵۳	..... ۲-۹-۳ RBF SVM با کاهش ابعاد و با نرمال سازی
۵۳	..... ۲-۹-۴ Logistic Regression با کاهش ابعاد و با نرمال سازی
۵۴	..... ۲-۹-۵ MLP با کاهش ابعاد و با نرمال سازی
۵۵	..... ۲-۹-۶ Ensemble Method با کاهش ابعاد و با نرمال سازی
۵۶	..... ۲-۹-۷ Linear SVM با کاهش ابعاد و بدون نرمال سازی
۵۷	..... ۲-۹-۸ Naïve Bayes با کاهش ابعاد و بدون نرمال سازی
۵۸	..... ۲-۹-۹ RBF SVM با کاهش ابعاد و بدون نرمال سازی
۵۹	..... ۲-۹-۱۰ Logistic Regression با کاهش ابعاد و بدون نرمال سازی
۶۰	..... ۲-۹-۱۱ MLP با کاهش ابعاد و بدون نرمال سازی
۶۰	..... ۲-۹-۱۲ Ensemble Method با کاهش ابعاد و بدون نرمال سازی

۶۱	۲-۹-نتیجه‌گیری و جمع‌بندی
۶۳	<b>۳-خوبه‌بندی</b>
۶۳	۳-۱-مقدمه
۶۳	۳-۲-K-means Algorithm
۶۴	۳-۳-پردازش داده‌ها
۶۶	۳-۴-پیاده‌سازی الگوریتم K-Means و اجرای شبیه‌سازی
۷۶	۳-۵-Spectral Clustering
۸۱	۳-۶-نتیجه‌گیری و جمع‌بندی
۸۲	<b>۴-پیاده‌سازی ASR (بخش امتیازی)</b>
۸۲	۴-۱-مقدمه
۸۲	۴-۲-پیش‌پردازش (Preprocessing)
۸۳	۴-۳-آموزش (Training)
۸۴	۴-۴-ارزیابی (Evaluation)
۸۵	۴-۵-نتیجه‌گیری
۸۹	<b>جمع‌بندی و نتیجه‌گیری</b>

## شکل‌ها

۱۱	شکل ۱. ویژگی‌های استخراج شده از داده‌های صوتی خام (بالا spectrogram و پایین MFCC)
۱۲	شکل ۲. ویژگی‌های استخراج شده بعد از silenc trimming (بالا spectrogram و پایین MFCC)
۱۳	شکل ۳. ویژگی‌های استخراج شده بعد از حذف نویز (بالا spectrogram و پایین MFCC)
۱۴	شکل ۴. توزیع جنسیت.
۱۶	شکل ۵. نمودار هیستوگرام برای توزیع برچسب‌ها
۲۶	شکل ۶. ماتریس درهمریختگی و منحنی ROC برای Linear SVM بدون کاهش بعد و با نرمال‌سازی
۲۶	شکل ۷. نتایج ارزیابی کلاسی و کلی داده برای Linear SVM بدون کاهش بعد و با نرمال‌سازی
۲۷	شکل ۸. ماتریس درهمریختگی و منحنی ROC برای Naive Bayes بدون کاهش بعد و با نرمال‌سازی
۲۷	شکل ۹. نتایج ارزیابی کلاسی و کلی داده برای Naive Bayes بدون کاهش بعد و با نرمال‌سازی
۲۸	شکل ۱۰. ماتریس درهمریختگی و منحنی ROC برای RBF SVM بدون کاهش بعد و با نرمال‌سازی
۲۸	شکل ۱۱. نتایج ارزیابی کلاسی و کلی داده برای RBF SVM بدون کاهش بعد و با نرمال‌سازی
۲۹	شکل ۱۲. ماتریس درهمریختگی و منحنی ROC برای Logistic Regression بدون کاهش بعد و با نرمال‌سازی
۲۹	شکل ۱۳. نتایج ارزیابی کلاسی و کلی داده برای Logistic Regression بدون کاهش بعد و با نرمال‌سازی
۳۰	شکل ۱۴. ماتریس درهمریختگی و منحنی ROC برای MLP بدون کاهش بعد و با نرمال‌سازی
۳۰	شکل ۱۵. نتایج ارزیابی کلاسی و کلی داده برای MLP بدون کاهش بعد و با نرمال‌سازی
۳۱	شکل ۱۶. ماتریس درهمریختگی و منحنی ROC برای Ensemble models بدون کاهش بعد و با نرمال‌سازی
۳۱	شکل ۱۷. نتایج ارزیابی کلاسی و کلی داده برای Ensemble Models بدون کاهش بعد و با نرمال‌سازی
۳۲	شکل ۱۸. ماتریس درهمریختگی و منحنی ROC برای Linear SVM بدون کاهش بعد و بدون نرمال‌سازی
۳۲	شکل ۱۹. نتایج ارزیابی کلاسی و کلی داده برای Linear SVM بدون کاهش بعد و بدون نرمال‌سازی
۳۳	شکل ۲۰. ماتریس درهمریختگی و منحنی ROC برای Naive Bayes بدون کاهش بعد و بدون نرمال‌سازی
۳۳	شکل ۲۱. نتایج ارزیابی کلاسی و کلی داده برای Naive Bayes بدون کاهش بعد و بدون نرمال‌سازی
۳۴	شکل ۲۲. ماتریس درهمریختگی و منحنی ROC برای RBF SVM بدون کاهش بعد و بدون نرمال‌سازی
۳۴	شکل ۲۳. نتایج ارزیابی کلاسی و کلی داده برای RBF SVM بدون کاهش بعد و بدون نرمال‌سازی
۳۵	شکل ۲۴. ماتریس درهمریختگی و منحنی ROC برای Logistic Regression بدون کاهش بعد و بدون نرمال‌سازی
۳۵	شکل ۲۵. نتایج ارزیابی کلاسی و کلی داده برای Logistic Regression بدون کاهش بعد و بدون نرمال‌سازی
۳۶	شکل ۲۶. ماتریس درهمریختگی و منحنی ROC برای MLP بدون کاهش بعد و بدون نرمال‌سازی
۳۶	شکل ۲۷. نتایج ارزیابی کلاسی و کلی داده برای MLP بدون کاهش بعد و بدون نرمال‌سازی
۳۷	شکل ۲۸. ماتریس درهمریختگی و منحنی ROC برای Ensemble models بدون کاهش بعد و بدون نرمال‌سازی
۳۸	شکل ۲۹. نتایج ارزیابی کلاسی و کلی داده برای Ensemble Models بدون کاهش بعد و بدون نرمال‌سازی
۳۸	شکل ۳۰. توجیه پراکندگی تجمعی با PCA
۳۹	شکل ۳۱. پراکندگی داده‌ها با دو ویژگی در PCA
۳۹	شکل ۳۲. ماتریس درهمریختگی و منحنی ROC برای Linear SVM با کاهش بعد PCA و با نرمال‌سازی
۴۰	شکل ۳۳. نتایج ارزیابی کلاسی و کلی داده برای Linear SVM با کاهش بعد PCA و با نرمال‌سازی
۴۰	شکل ۳۴. ماتریس درهمریختگی و منحنی ROC برای Naive Bayes با کاهش بعد PCA و با نرمال‌سازی
۴۱	شکل ۳۵. نتایج ارزیابی کلاسی و کلی داده برای Naive Bayes با کاهش بعد PCA و با نرمال‌سازی

۴۱. شکل ۳۶. ماتریس درهمریختگی و منحنی ROC برای RBF SVM با کاهش بعد PCA و با نرمال سازی

۴۲. شکل ۳۷. نتایج ارزیابی کلاسی و کلی داده برای RBF SVM با کاهش بعد PCA و با نرمال سازی

۴۲. شکل ۳۸. ماتریس درهمریختگی و منحنی ROC برای Logistic Regression با کاهش بعد PCA و با نرمال سازی

۴۳. شکل ۳۹. نتایج ارزیابی کلاسی و کلی داده برای Logistic Regression با کاهش بعد PCA و با نرمال سازی

۴۰. شکل ۴۰. ماتریس درهمریختگی و منحنی ROC برای MLP با کاهش بعد PCA و با نرمال سازی

۴۱. شکل ۴۱. نتایج ارزیابی کلاسی و کلی داده برای MLP با کاهش بعد PCA و با نرمال سازی

۴۲. شکل ۴۲. ماتریس درهمریختگی و منحنی ROC برای Ensemble models با کاهش بعد PCA و با نرمال سازی

۴۳. شکل ۴۳. نتایج ارزیابی کلاسی و کلی داده برای Ensemble Models با کاهش بعد PCA و با نرمال سازی

۴۴. شکل ۴۴. ماتریس درهمریختگی و منحنی ROC برای Linear SVM با کاهش بعد PCA و بدون نرمال سازی

۴۵. شکل ۴۵. نتایج ارزیابی کلاسی و کلی داده برای Linear SVM با کاهش بعد PCA و بدون نرمال سازی

۴۶. شکل ۴۶. ماتریس درهمریختگی و منحنی ROC برای Naive Bayes با کاهش بعد PCA و بدون نرمال سازی

۴۷. شکل ۴۷. نتایج ارزیابی کلاسی و کلی داده برای Naive Bayes با کاهش بعد PCA و بدون نرمال سازی

۴۸. شکل ۴۸. ماتریس درهمریختگی و منحنی ROC برای RBF SVM با کاهش بعد PCA و بدون نرمال سازی

۴۹. شکل ۴۹. نتایج ارزیابی کلاسی و کلی داده برای RBF SVM با کاهش بعد PCA و بدون نرمال سازی

۵۰. شکل ۵۰. ماتریس درهمریختگی و منحنی ROC برای Logistic Regression با کاهش بعد PCA و بدون نرمال سازی

۵۱. شکل ۵۱. نتایج ارزیابی کلاسی و کلی داده برای Logistic Regression با کاهش بعد PCA و بدون نرمال سازی

۵۲. شکل ۵۲. ماتریس درهمریختگی و منحنی ROC برای MLP با کاهش بعد PCA و بدون نرمال سازی

۵۳. شکل ۵۳. نتایج ارزیابی کلاسی و کلی داده برای MLP با کاهش بعد PCA و بدون نرمال سازی

۵۴. شکل ۵۴. ماتریس درهمریختگی و منحنی ROC برای Ensemble models با کاهش بعد PCA و بدون نرمال سازی

۵۵. شکل ۵۵. نتایج ارزیابی کلاسی و کلی داده برای Ensemble Models با کاهش بعد PCA و بدون نرمال سازی

۵۶. شکل ۵۶. ماتریس درهمریختگی و منحنی ROC برای Linear SVM با کاهش بعد LDA و با نرمال سازی

۵۷. شکل ۵۷. نتایج ارزیابی کلاسی و کلی داده برای Linear SVM با کاهش بعد LDA و با نرمال سازی

۵۸. شکل ۵۸. ماتریس درهمریختگی و منحنی ROC برای Naive Bayes با کاهش بعد LDA و با نرمال سازی

۵۹. شکل ۵۹. نتایج ارزیابی کلاسی و کلی داده برای Naive Bayes با کاهش بعد LDA و با نرمال سازی

۶۰. شکل ۶۰. ماتریس درهمریختگی و منحنی ROC برای RBF SVM با کاهش بعد LDA و با نرمال سازی

۶۱. شکل ۶۱. نتایج ارزیابی کلاسی و کلی داده برای RBF SVM با کاهش بعد LDA و با نرمال سازی

۶۲. شکل ۶۲. ماتریس درهمریختگی و منحنی ROC برای Logistic Regression با کاهش بعد LDA و با نرمال سازی

۶۳. شکل ۶۳. نتایج ارزیابی کلاسی و کلی داده برای Logistic Regression با کاهش بعد LDA و با نرمال سازی

۶۴. شکل ۶۴. ماتریس درهمریختگی و منحنی ROC برای MLP با کاهش بعد LDA و با نرمال سازی

۶۵. شکل ۶۵. نتایج ارزیابی کلاسی و کلی داده برای MLP با کاهش بعد LDA و با نرمال سازی

۶۶. شکل ۶۶. ماتریس درهمریختگی و منحنی ROC برای Ensemble models با کاهش بعد LDA و با نرمال سازی

۶۷. شکل ۶۷. نتایج ارزیابی کلاسی و کلی داده برای Ensemble Models با کاهش بعد LDA و با نرمال سازی

۶۸. شکل ۶۸. ماتریس درهمریختگی و منحنی ROC برای Linear SVM با کاهش بعد LDA و بدون نرمال سازی

۶۹. شکل ۶۹. نتایج ارزیابی کلاسی و کلی داده برای Linear SVM با کاهش بعد LDA و بدون نرمال سازی

۷۰. شکل ۷۰. ماتریس درهمریختگی و منحنی ROC برای Naive Bayes با کاهش بعد LDA و بدون نرمال سازی

۷۱. شکل ۷۱. نتایج ارزیابی کلاسی و کلی داده برای Naive Bayes با کاهش بعد LDA و بدون نرمال سازی

۷۲. شکل ۷۲. ماتریس درهمریختگی و منحنی ROC برای RBF SVM با کاهش بعد LDA و بدون نرمال سازی

شکل ۷۳. نتایج ارزیابی کلاسی و کلی داده برای RBF SVM با کاهش بعد LDA و بدون نرمال سازی ..... ۵۸
شکل ۷۴. ماتریس درهم ریختگی و منحنی ROC برای Logistic Regression با کاهش بعد LDA و بدون نرمال سازی ..... ۵۹
شکل ۷۵. نتایج ارزیابی کلاسی و کلی داده برای Logistic Regression با کاهش بعد LDA و بدون نرمال سازی ..... ۵۹
شکل ۷۶. ماتریس درهم ریختگی و منحنی ROC برای MLP با کاهش بعد LDA و بدون نرمال سازی ..... ۶۰
شکل ۷۷. نتایج ارزیابی کلاسی و کلی داده برای MLP با کاهش بعد LDA و بدون نرمال سازی ..... ۶۰
شکل ۷۸. ماتریس درهم ریختگی و منحنی ROC برای Ensemble models با کاهش بعد LDA و بدون نرمال سازی ..... ۶۰
شکل ۷۹. نتایج ارزیابی کلاسی و کلی داده برای Ensemble Models با کاهش بعد LDA و بدون نرمال سازی ..... ۶۱
شکل ۸۰. محاسبه مقادیر silhouette score برای تعداد کلاسترها در الگوریتم K-means ..... ۶۷
شکل ۸۱. آنالیز silhouette در الگوریتم K-means با تعداد کلاستر ۲ ..... ۷۰
شکل ۸۲. آنالیز silhouette در الگوریتم K-means با تعداد کلاستر ۴ ..... ۷۱
شکل ۸۳. آنالیز silhouette در الگوریتم K-means با تعداد کلاستر ۶ ..... ۷۱
شکل ۸۴. آنالیز silhouette در الگوریتم K-means با تعداد کلاستر ۸ ..... ۷۲
شکل ۸۵. آنالیز PCA در دو بعد با استفاده از Scatter plot ..... ۷۳
شکل ۸۶. خوش‌های یافته شده از الگوریتم K-means با ۲ کلاستر و لیبل‌ها قبل از خوش‌بندی ..... ۷۴
شکل ۸۷. خوش‌های یافته شده از الگوریتم K-means با ۴ کلاستر و لیبل‌ها قبل از خوش‌بندی ..... ۷۴
شکل ۸۸. خوش‌های یافته شده از الگوریتم K-means با ۶ کلاستر و لیبل‌ها قبل از خوش‌بندی ..... ۷۵
شکل ۸۹. خوش‌های یافته شده از الگوریتم K-means با ۸ کلاستر و لیبل‌ها قبل از خوش‌بندی ..... ۷۵
شکل ۹۰. محاسبه مقادیر silhouette score برای تعداد کلاسترها با روش spectral clustering ..... ۷۶
شکل ۹۱. آنالیز silhouette در روش spectral clustering با تعداد کلاستر ۲ ..... ۷۷
شکل ۹۲. آنالیز silhouette در روش spectral clustering با تعداد کلاستر ۴ ..... ۷۷
شکل ۹۳. آنالیز silhouette در روش spectral clustering با تعداد کلاستر ۶ ..... ۷۸
شکل ۹۴. آنالیز silhouette در روش spectral clustering با تعداد کلاستر ۸ ..... ۷۸
شکل ۹۵. خوش‌های یافته شده از روش Spectral clustering با ۲ کلاستر و لیبل‌ها قبل از خوش‌بندی با کاهش بعد ..... ۷۹
شکل ۹۶. خوش‌های یافته شده از روش Spectral clustering با ۴ کلاستر و لیبل‌ها قبل از خوش‌بندی با کاهش بعد ..... ۷۹
شکل ۹۷. خوش‌های یافته شده از روش Spectral clustering با ۶ کلاستر و لیبل‌ها قبل از خوش‌بندی با کاهش بعد ..... ۸۰
شکل ۹۸. خوش‌های یافته شده از روش Spectral clustering با ۸ کلاستر و لیبل‌ها قبل از خوش‌بندی با کاهش بعد ..... ۸۰
شکل ۹۹. نمودار تغییرات خطأ در طول آموزش ..... ۸۴
شکل ۱۰۰. نمودار تغییرات WER در طول آموزش ..... ۸۵
شکل ۱۰۱. نمودار پراکندگی WER بر اساس جنسیت ..... ۸۵
شکل ۱۰۲. نمودار پراکندگی WER بر اساس لهجه ..... ۸۶
شکل ۱۰۳. نمودار پراکندگی WER بر اساس لحن ..... ۸۶
شکل ۱۰۴. نمودار توزیع ویژگی‌های جنسیت، لهجه و لحن ..... ۸۷
شکل ۱۰۵. محاسبه همبستگی ویژگی‌های جنسیت، لهجه و لحن با WER ..... ۸۷
شکل ۱۰۶. ماتریس همبستگی (متد spearman) ..... ۸۸
شکل ۱۰۷. ماتریس همبستگی (متد pearson) ..... ۸۹

## جدول‌ها

- جدول ۱. خلاصه نتایج و مقایسه مدل‌ها با کاهش ابعاد و بدون کاهش ابعاد و همراه با نرمال‌سازی و عدم نرمال‌سازی ..... ۶۱  
۸۳ ..... جدول ۲. موارد استفاده شده در نرمال‌سازی توسط کتابخانه hazm

## مقدمه

در این گزارش ابتدا مراحلی که در پیش‌پردازش و تمیز کردن داده‌ها انجام دادیم را ذکر کرده و در ادامه روندی را برای طراحی طبقه‌بندی‌های مختلف طی می‌نماییم و در انتها دقیق مدل‌های مختلفی که آموزش دادیم را با یکدیگر مقایسه خواهیم نمود و سپس خوشبندی را برای داده‌ها انجام خواهیم داد. در بخش انتهایی نیز ASR را انجام داده و نتایج آن را تحلیل خواهیم نمود.

### ۱-پیش‌پردازش داده‌ها

پیش‌پردازش و تمیز کردن داده‌های صوتی با توجه به نوع داده‌ها و تسلیک مورد نظر می‌تواند شامل مراحل مختلفی باشد. در این بخش ابتدا به تمیز کردن داده‌های صوتی خواهیم پرداخت که شامل مراحل زیر می‌باشند.

- یکسان کردن برچسب‌ها

در دیتاست داده شده برخی برچسب‌ها به شکل‌های مختلف نوشته شده بودند که همگی آن‌ها یکسان‌سازی شدند.

```
before: ['فارسی', 'ترکی', 'خراسانی', 'بزدی', 'farsi', 'فارسی', 'شیرازی']  
after: ['فارسی', 'شیرازی', 'ترکی', 'خراسانی', 'بزدی']
```

```
before: ['question', 'normal', 'imperative', 'incomplete', 'exclamatory', 'normal',  
         'question', 'imperative', 'exclamatory', 'incomplete', 'Normal', 'Question',  
         'nomal', 'question/incomplete', 'incomplete', 'exclamative', 'quenstion',  
         'nortmal', 'impreative']  
after: ['question', 'normal', 'imperative', 'incomplete', 'exclamatory']
```

- یکسان کردن sample rate فایل‌های صوتی

اغلب در پردازش گفتار از فرکانس‌های ۱۶ یا ۲۲,۰۵ کیلوهرتز استفاده می‌شود، زیرا گفتار انسان معمولاً دارای فرکانس‌های بسیار پایین تر از آنچه موسیقی در بر می‌گیرد می‌باشد. این نرخ‌های پایین تر برای گرفتن وضوح گفتار و در عین حال قابل کنترل نگه داشتن اندازه داده کافی است. ما برای این داده‌های فرکانس ۱۶ کیلوهرتز را در نظر گرفتیم.

- حذف نویز صدا با استفاده از کتابخانه noisereduce

از این کتابخانه جهت حذف نویز استفاده نمودیم.

- **تشخیص و حذف فضای خالی(Skot)**

در ابتدا و انتهای فایل های صوتی (silence trimming) با استفاده از کتابخانه PyDub. بعد از اینکه silence trimming انجام شد فایل جدید را مجدد ذخیره میکنیم. (در این بین برخی از فایل های صوتی دارای محتوای خالی بودند که در طی این فرایند حذف شدند)

پس از تمیز کردن داده های صوتی، ویژگی هایی که در طبقه بندی و خوش بندی به آنها نیاز داریم را طبق موارد زیر استخراج می نماییم.

- **Spectrogram**

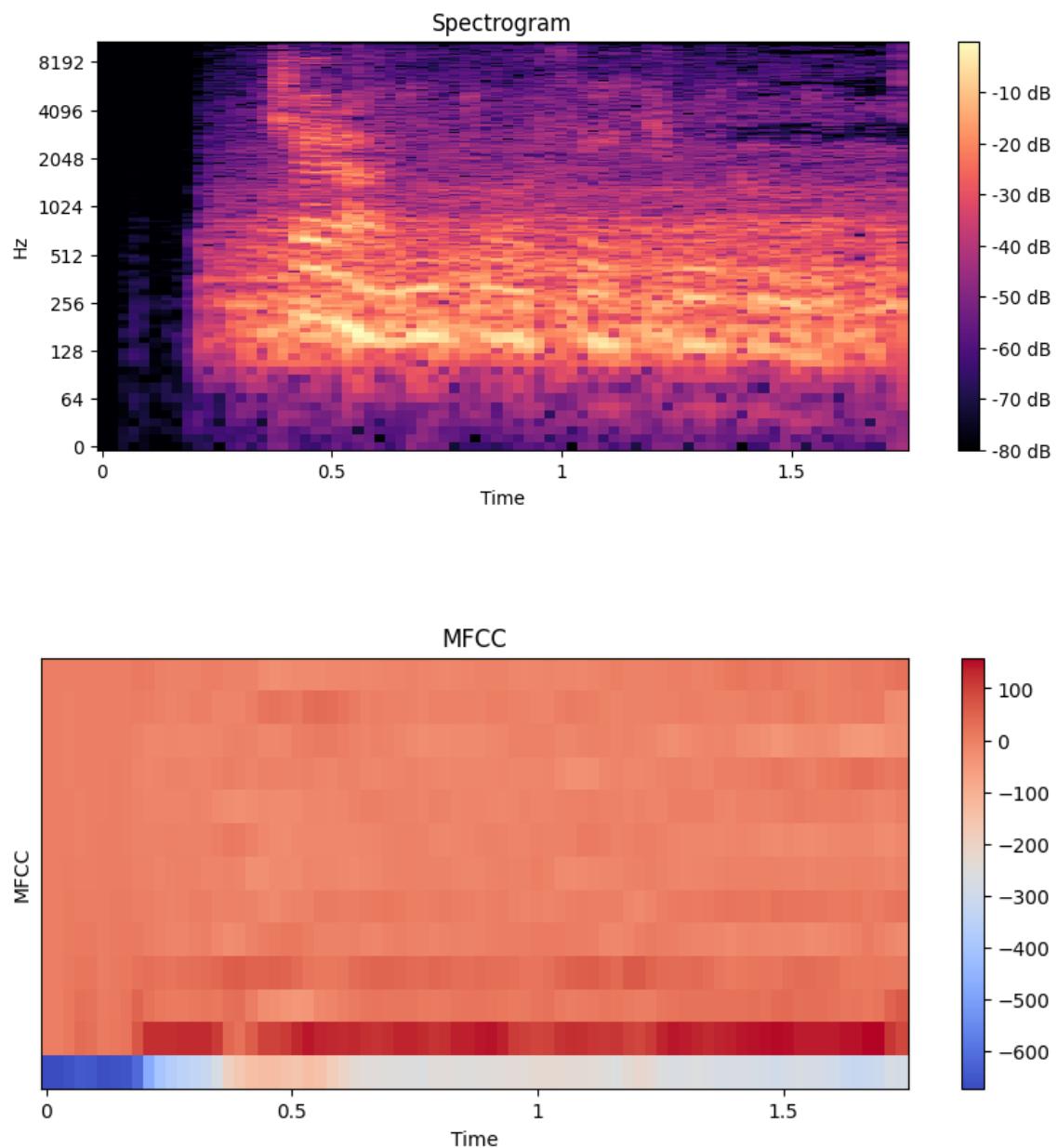
طیف فرکانس های صوتی و تغییرات آن در طول بازه زمانی را نشان خواهد داد که از آن استفاده می کنیم.

- **:MFCCs ضرایب**

ضرایب MFCC یا Mel-Frequency Cepstral Coefficients بر مبنای سیستم شنوایی انسان برای یک سیگنال صوتی انجام می شود. هر فریم سیگنال ابتدا در پنجره همینگ ضرب می شود و سپس از نتیجه تبدیل فوریه گسسته گرفته می شود. سپس طیف سیگنال از تعداد دلخواهی عبورداده می شود. این فیلترها تفکیک فرکانسی سیستم ادراک گوش انسان را شبیه سازی می کنند. این کار به طور موثر توزیع توان در مقیاس Mel را میگیرد، بنابراین نمایش فشرده ای از صدا را ارائه می دهد. این باعث می شود که MFCC در کاربردهای مختلفی مانند تشخیص صدا، تشخیص ابزار موسیقی و به طور کلی در هر سیستمی که صدا و ویژگی های آن اهمیت دارد بسیار مفید باشد. در اکثر پژوهش های مرتبط با صوت تنها از ۱۲ یا ۱۳ ضریب ابتدایی MFC استفاده می شود. ما نیز در این مسئله  $n_{mfcc}$  را برابر ۱۳ گرفتیم.

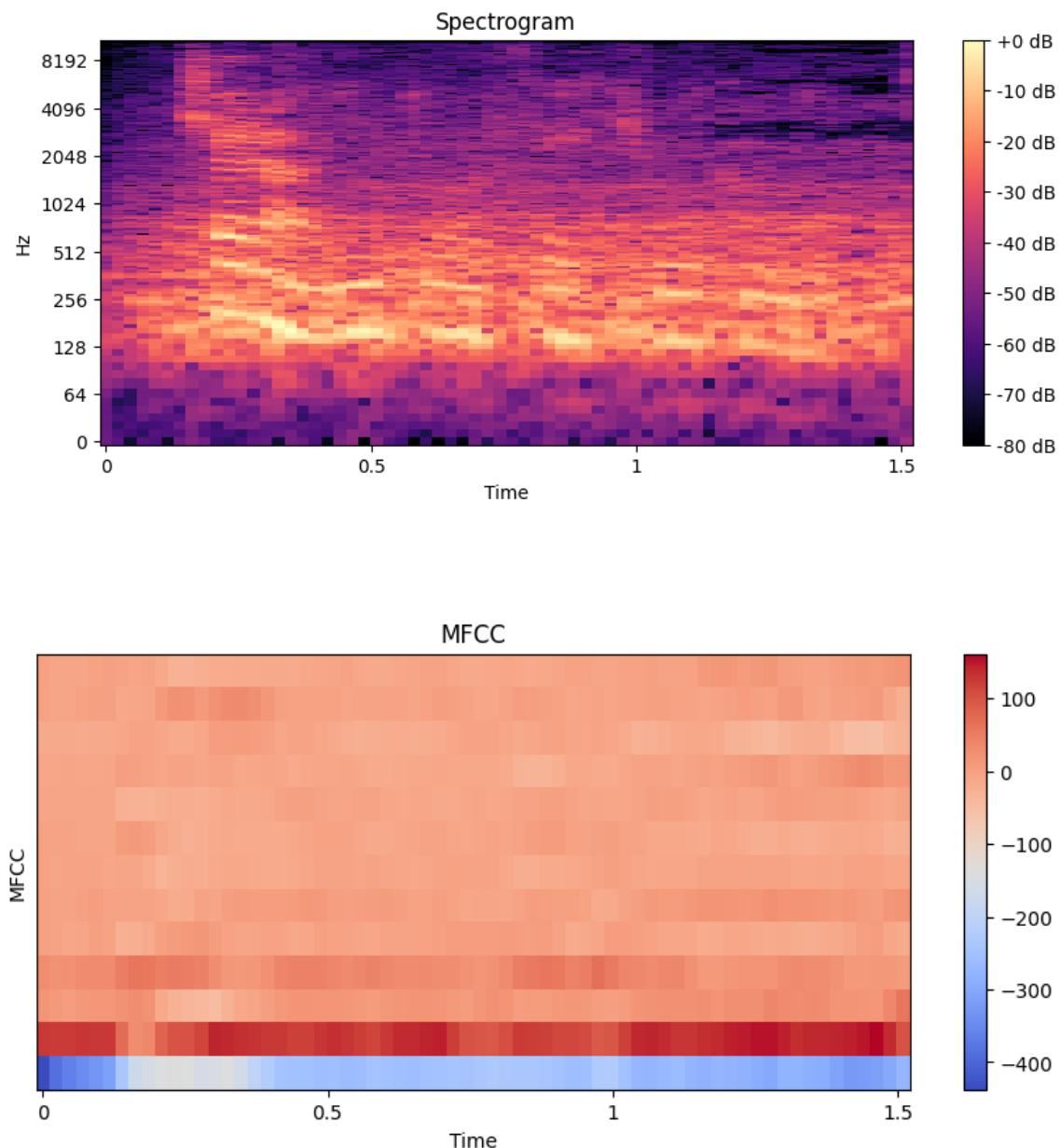
خروجی استفاده از این دو ویژگی در طول تمیز کردن داده را در ادامه با نشان دادن شکل نمایش خواهیم داد.

## نمونه ضرایب داده های خام اولیه



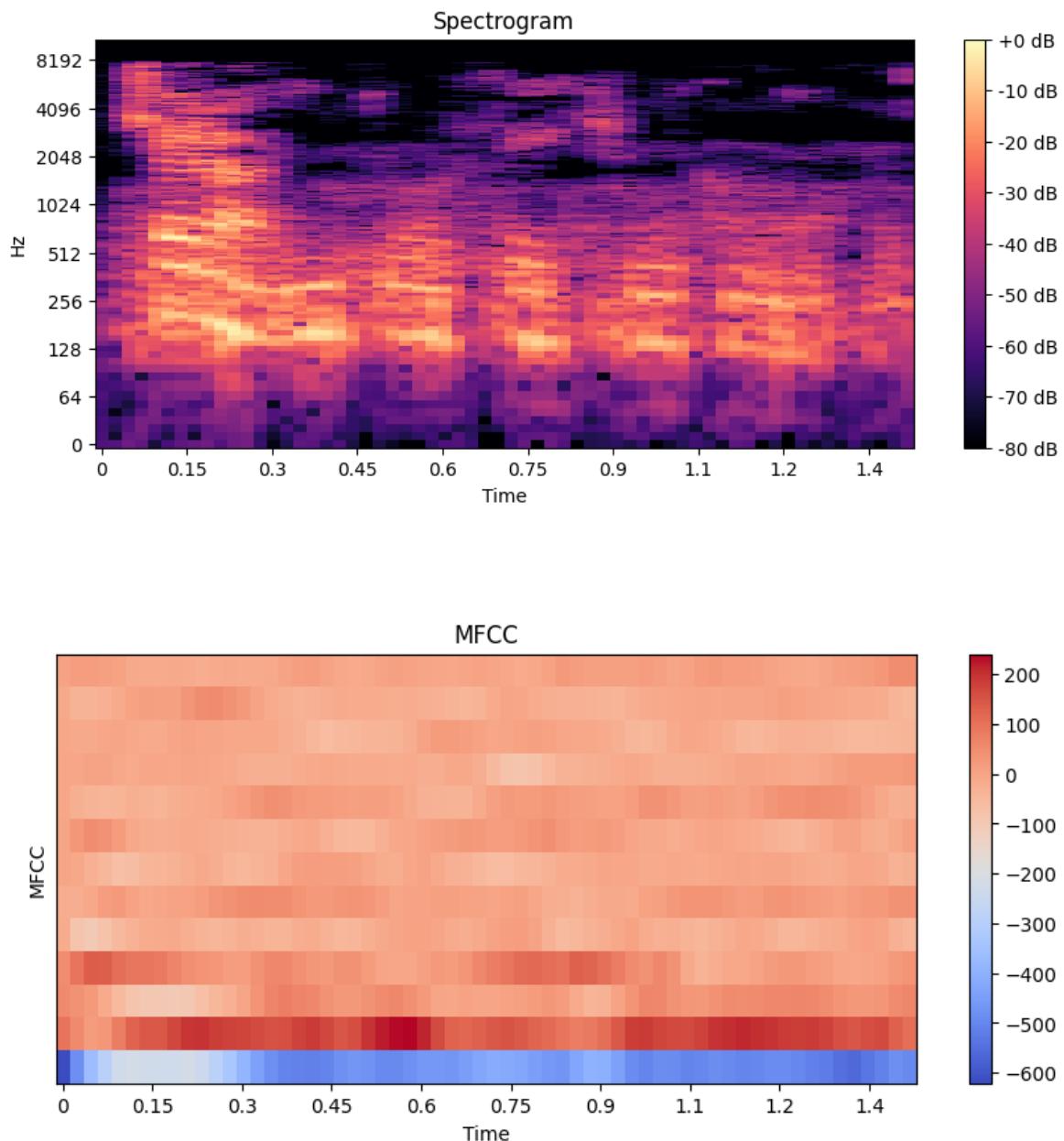
شکل ۱. ویژگی های استخراج شده از داده های صوتی خام (بالا spectrogram و پایین MFCC)

## ویژگی داده بعد از silenc trimming



شکل ۲. ویژگی های استخراج شده بعد از silenc trimming (بالا spectrogram و پایین MFCC)

## ویژگی داده ها بعد از noise filtering



شکل ۳. ویژگی های استخراج شده بعد از حذف نویز (بالا spectrogram و پایین MFCC)

در انتهای دیتابست جدید به همراه ویژگی های استخراج شده و همچنین فایل های صوتی تمیز شده را در آکانت درایو ذخیره نموده تا در بخش های بعدی از آن ها استفاده نماییم.

## ۲-طبقه‌بندی

### ۲-۱-مقدمه

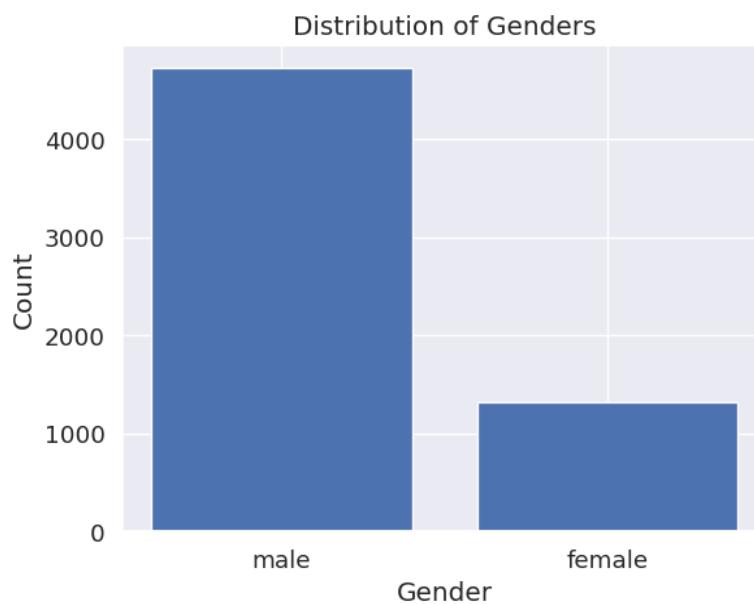
در این قسمت قصد داریم تا با استفاده از فایل‌های صوتی تمیز شده و ویژگی‌های استخراج شده، عملیات طبقه‌بندی بر روی داده‌ها را انجام دهیم. به همین منظور با استفاده از قطعه کد زیر، از ویژگی‌های استخراج شده استفاده نموده و سراغ بخش‌های بعدی خواهیم رفت.

```
▶ import numpy as np
import pandas as pd
# Specify the path to your NPZ file
#file_path = '/content/drive/MyDrive/ML/project/mfccs.npz'
file_path = '/content/drive/MyDrive/ML_Project/mfccs.npz'

data = np.load(file_path, allow_pickle=True)
```

### ۲-۲-بررسی آماری داده‌ها

ابتدا توزیع‌های آماری، جهت اشراف بیشتر بر روی داده‌ها را به دست خواهیم آورد. این کار به طبقه‌بندی کمک خواهد نمود. به همین منظور توزیع جنسیت در دیتاست را مورد بررسی قرار می‌دهیم که در شکل ۴ آورده شده است.



شکل ۴. توزیع جنسیت

شکل ۴ نشان‌دهنده توزیع جنسیت‌ها در یک داده‌نما است که بر اساس آن، تعداد افراد مرد بسیار بیشتر از تعداد افراد زن است. بر اساس اطلاعات موجود در محور عمودی (Count)، تعداد افراد مرد بیش از ۴۰۰۰ نفر و تعداد افراد زن کمتر از ۲۰۰۰ نفر می‌باشد. در ادامه با استفاده از کد زیر داده‌ها را به یک اندازه می‌کنیم.

```
▶ import numpy as np
from collections import Counter
from imblearn.under_sampling import RandomUnderSampler

# Count the class distribution before undersampling
class_distribution_before = Counter(y)
print("Class distribution before undersampling:", class_distribution_before)

# Calculate the desired number of samples for the majority class
minority_class = min(class_distribution_before, key=class_distribution_before.get)
majority_class = "male"
desired_majority_samples = class_distribution_before[minority_class]

# Create the RandomUnderSampler with the desired sampling strategy
undersampler = RandomUnderSampler(sampling_strategy={majority_class: desired_majority_samples})

# Apply the undersampling to the dataset
X_resampled, y_resampled = undersampler.fit_resample(features, y)

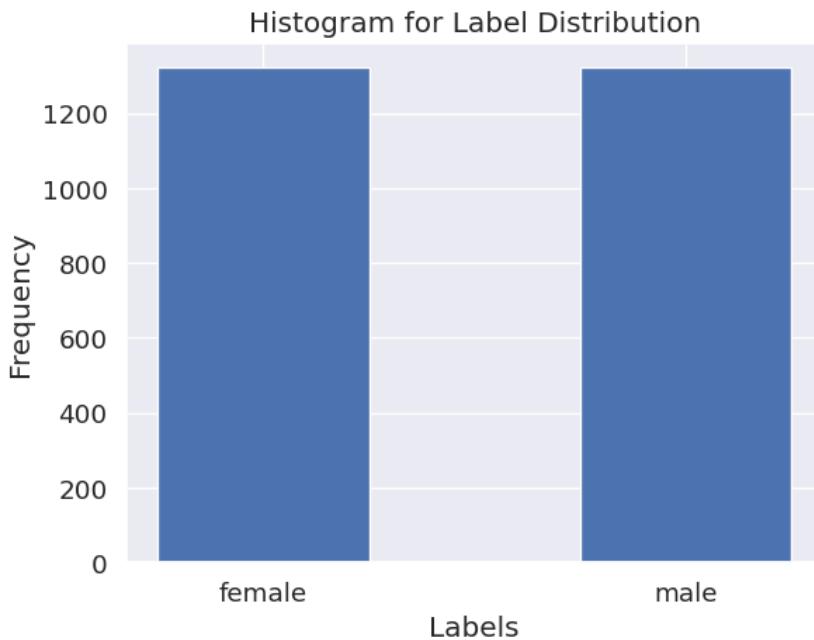
# Count the class distribution after undersampling
class_distribution_after = Counter(y_resampled)
print("Class distribution after undersampling:", class_distribution_after)
```

⌚ Class distribution before undersampling: Counter({'male': 4720, 'female': 1322})
⌚ Class distribution after undersampling: Counter({'female': 1322, 'male': 1322})

این کد پایتون یک نمونه از استفاده از کتابخانه `imblearn` برای انجام under sampling در داده‌های طبقه‌بندی نامتعادل است. Under sampling روشی است برای کاهش تعداد نمونه‌ها در کلاس اکثریت به منظور دستیابی به توازن بین کلاس‌های مختلف در یک مجموعه داده. در این مورد خاص، کد برای متعادل کردن تعداد نمونه‌ها بین دو کلاس جنسیتی مرد و زن استفاده شده است.

این کد به اطمینان از اینکه کلاس‌های جنسیتی در مجموعه داده دارای تعداد نمونه‌های برابری هستند کمک می‌کند، که می‌تواند به رفع بیاس مدل یادگیری ماشین در زمان آموزش کمک کند.

شکل ۵ نشان‌دهنده توزیع متعادل برچسب‌ها (Labels) برای دو گروه جنسیتی زن (female) و مرد (male) پس از اعمال under sampling است. هر دو ستون دارای ارتفاع برابری هستند، که نشان می‌دهد تعداد نمونه‌ها در هر دو گروه جنسیتی مشابه است. این یعنی کد `undersampling` که پیشتر نشان کردیم، به درستی کار کرده و تعداد نمونه‌ها در کلاس اکثریت (که در این مورد مردان بودند) را به تعداد نمونه‌های کلاس اقلیت (زنان) کاهش داده است.



شکل ۵. نمودار هیستوگرام برای توزیع برچسبها

این نتیجه مطلوب برای آموزش مدل‌های یادگیری ماشین است، زیرا باعث می‌شود که مدل داده‌ها را بدون بایاس ناشی از توزیع ناهمگون کلاس‌ها یاد بگیرد. توزیع متعادل شده می‌تواند به بهبود عملکرد مدل در طول آموزش و هنگام تست بر روی داده‌های جدید کمک کند.

### ۲-۳- تقسیم داده‌های آموزش و تست

ما از کتابخانه `sklearn.model_selection` داده‌های موجود را به دو مجموعه تقسیم می‌کند استفاده کردیم نمودیم که شامل مجموعه داده‌های آموزشی (که برای آموزش مدل استفاده می‌شوند) و مجموعه داده‌های تست (که برای ارزیابی عملکرد مدل استفاده می‌شوند). این تقسیم‌بندی مطابق کد زیر انجام شد.

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Split your data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_resampled, y_resampled, test_size=0.25, random_state=42)
```

از ۷۵٪ داده‌ها برای تست و ۲۵٪ باقی‌مانده برای آموزش استفاده می‌شوند. هم این مقدار اطمینان می‌دهد که تقسیم‌بندی داده‌ها به صورت قابل تکرار است، به این معنی که هر بار اجرای کد، تقسیم‌بندی یکسانی انجام می‌گیرد.

### ۲-۴- روش‌های طبقه‌بندی

برای طبقه‌بندی از چند روش مختلف استفاده نمودیم که ابتدا آن‌ها را به اختصار توضیح خواهیم داد.

## Linear SVM-۲-۴-۱

روش ماشین بردار پشتیبانی خطی (Linear SVM) یک الگوریتم طبقه‌بندی است که در یادگیری ماشین برای تشخیص دو کلاس یا بیشتر به کار می‌رود. در حالت خطی آن، هدف این است که یک خط، صفحه یا هایپرپلین (در فضاهای با بعد بالا) را پیدا کنیم که دو کلاس را با حداقل فاصله از هم جدا کند. این فاصله به عنوان "مارجین" شناخته می‌شود و ماشین بردار پشتیبانی خطی سعی دارد تا مارجین را به حداقل برساند. در این میان، بردارهای پشتیبان نقاط داده‌ای هستند که نزدیک‌ترین فاصله را به خط جداکننده دارند و در تعیین موقعیت و شیب این خط نقش اساسی بازی می‌کنند.

برای مجموعه داده‌هایی که به صورت خطی قابل جداسازی هستند مناسب است، یعنی وقتی که می‌توان بدون خطا یک خط یا صفحه را به گونه‌ای رسم کرد که همه نمونه‌های یک کلاس در یک طرف و نمونه‌های کلاس دیگر در طرف دیگر قرار گیرند. اگر داده‌ها به صورت خطی قابل جداسازی نباشند، می‌توان از روش‌های مختلفی مانند افزودن ویژگی‌های جدید یا استفاده از هسته‌ها (کرنل‌ها) برای افزایش بعد داده‌ها و یافتن جداسازی بهینه استفاده کرد. اما در مواردی که داده‌ها به طور خطی قابل جداسازی نیستند، استفاده از Linear SVM ممکن است نتایج کمتر مطلوبی داشته باشد.

## Naïve Bayes-۲-۴-۲

بیز ساده انگارانه (Naive Bayes) یک الگوریتم طبقه‌بندی است که بر پایه قانون بیز کار می‌کند و با فرض استقلال شرطی نایو (ساده‌لوحانه) بین ویژگی‌ها به کار گرفته می‌شود. این الگوریتم در یادگیری ماشین برای پیش‌بینی احتمال وابستگی یک نمونه به یک کلاس خاص بر اساس ویژگی‌های آن نمونه استفاده می‌شود.

در اینجا چگونگی کار الگوریتم Naive Bayes را بررسی می‌کنیم:

۱. قانون بیز: این قانون یک روش ریاضی برای محاسبه احتمال یک رویداد با توجه به شواهد (یا اطلاعات) موجود است. فرمول این قانون به صورت زیر است:

$$P(A|B) = P(B|A)/P(A) P(B)$$

که در آن  $P(A|B)$  احتمال وقوع رویداد A با توجه به وقوع رویداد B است.

۲. استقلال شرطی: Naive Bayes با فرض استقلال شرطی بین ویژگی‌های مختلف عمل می‌کند، به این معنی که وجود یا عدم وجود یک ویژگی خاص تأثیری بر وجود یا عدم وجود ویژگی دیگری ندارد.

۳. مدل سازی و پیش بینی: الگوریتم با استفاده از داده های آموزشی، احتمالات مربوط به ویژگی ها را برای هر کلاس محاسبه می کند. سپس، برای پیش بینی کلاس یک نمونه جدید، احتمالات را با استفاده از ویژگی های آن نمونه و فرض استقلال شرطی به کار می برد تا کلاسی که بیشترین احتمال را دارد تعیین شود.

الگوریتم Naive Bayes سریع، ساده و اغلب کارآمد است، به ویژه در داده هایی با ابعاد بالا مانند متن کاوی یا تشخیص اسپم. با این حال، به دلیل فرض استقلال نایاب، ممکن است در مواردی که ویژگی ها واقعاً مستقل نیستند، دقت کمتری داشته باشد. با این وجود، حتی با این فرض ساده‌انگارانه، Naive Bayes می تواند عملکرد خوبی داشته باشد و اغلب به عنوان یک خط مبنا در بسیاری از مسائل طبقه‌بندی استفاده می شود.

### RBF SVM-۲-۴-۳

ماشین بردار پشتیبانی باتابع هسته‌ای شعاعی پایه (RBF SVM) یک نوع از الگوریتم SVM است که از یک تابع هسته‌ای شعاعی پایه (Radial Basis Function - RBF) برای تبدیل فضای ویژگی به یک فضای با بعد بالاتر استفاده می کند، جایی که امکان جداسازی داده ها با یک هایپرپلین خطی فراهم می شود.

اصول اساسی RBF SVM عبارتند از:

۱. تابع هسته‌ی RBF: این تابع معمولاً به صورت  $\exp(-\gamma ||x - x'||^2)$  نوشته می شود، که در آن یک پارامتر تنظیمی است که تعیین می کند تأثیر یک نمونه تا چه اندازه ای به دیگر نمونه ها می رسد، یعنی چقدر وسیع یک تک نمونه در فضای ویژگی تأثیر می گذارد  $x$  و  $x'$  نمایانگر نمونه های داده هستند.

۲. فضای ویژگی با بعد بالا RBF: به داده ها اجازه می دهد که به فضایی با بعد بالاتر منتقل شوند که در آن می توانند به صورت خطی قابل جداسازی باشند، حتی اگر در فضای اصلی قابل جداسازی نباشند.

۳. مارجین و بردارهای پشتیبان: مانند هر SVM دیگری، RBF SVM به دنبال یک هایپرپلین است که کلاس ها را با بیشترین مارجین ممکن از یکدیگر جدا می کند. نقاط داده ای که نزدیک ترین فاصله را به این هایپرپلین دارند به عنوان بردارهای پشتیبان شناخته می شوند.

۴. overfitting و تنظیم پارامترها: پارامتر  $\gamma$  و همچنین پارامتر جرمیمه  $C$  (که تعیین می کند چقدر از نمونه های اشتباه در آموزش جرمیمه می شوند) باید با دقت تنظیم شوند تا از overfit (برازش بیش از حد)

جلوگیری شود. این می‌تواند از طریق روش‌هایی مانند جستجوی شبکه‌ای یا اعتبارسنجی متقابل انجام شود.

RBF SVM به خصوص در مواردی که روابط بین کلاس‌ها و ویژگی‌ها پیچیده و غیرخطی است، مفید است. این الگوریتم قدرت بالایی در پیدا کردن جدایی‌های پیچیده در داده‌ها دارد، اما به دلیل نیاز به تنظیم دقیق پارامترها و محاسبات بیشتر نسبت به SVM خطی، ممکن است در مجموعه‌های داده‌ی بسیار بزرگ کمی کندر باشد.

#### Logistic Regressions-۲-۴-۴

رگرسیون لجستیک یک الگوریتم طبقه‌بندی معروف در یادگیری ماشین است که برای پیش‌بینی احتمال وقوع یک رویداد باینری (دو کلاسی) بر اساس یک یا چند متغیر مستقل استفاده می‌شود. این رویداد می‌تواند دو حالت داشته باشد، مانند موفقیت/شکست، بیمار/سالم، و غیره.

اصول اساسی رگرسیون لجستیک عبارتند از:

۱. مدل سازی احتمال: رگرسیون لجستیک مدلی را فرموله می‌کند که احتمال وقوع رویداد موفقیت (معمولًاً نمایش داده شده با  $\lambda^1$ ) را به عنوان یک تابع لجستیک از متغیرهای مستقل (ویژگی‌ها) تخمین می‌زند.

۲. تابع لینک: تابع لجستیک، که به صورت

$$\frac{1}{1+e^{-(\beta_0 + \beta_1 x_1 + \dots + \beta_k x_k)}}$$

نوشته می‌شود، تابع لینک در رگرسیون لجستیک است. این تابع احتمالات را به یک مقدار بین ۰ و ۱ تبدیل می‌کند.

۳. تخمین پارامترها: پارامترهای  $\beta$  مدل با استفاده از روش حداقل احتمال (Maximum Likelihood) تخمین زده می‌شوند. MLE (Estimation - MLE) سعی می‌کند پارامترهایی را پیدا کند که احتمال داده‌های مشاهده شده را به حداقل برساند.

۴. تصمیم‌گیری: برای تعیین کلاس یک نمونه، احتمال محاسبه شده با یک آستانه (معمولًاً ۰,۵) مقایسه می‌شود: اگر احتمال بیشتر از آستانه باشد، نمونه به کلاس مثبت داده می‌شود؛ در غیر این صورت، به کلاس منفی نسبت داده می‌شود.

رگرسیون لجستیک به دلیل سادگی، شفافیت تفسیر، و کارایی در مواردی که رابطه بین متغیرهای مستقل و رویداد مورد نظر لجستیکی است، محبوبیت دارد. این الگوریتم بیشتر در مسائل طبقه‌بندی با دو کلاس استفاده می‌شود، اما می‌توان آن را با اندکی تغییر برای مسائل چند کلاسی نیز به کار برد.

#### MLP-۲-۴-۵

شبکه عصبی چند لایه (MLP)، که به نام شبکه عصبی پیشرونده چند لایه نیز شناخته می‌شود، یک مدل یادگیری عمیق است که از چندین لایه نورون‌ها تشکیل شده و می‌تواند الگوهای پیچیده‌تری را نسبت به الگوریتم‌های طبقه‌بندی سنتی مانند رگرسیون لجستیک یا SVM شناسایی کند. هر نورون در یک لایه با تمام نورون‌های لایه بعدی متصل است و داده‌ها از ورودی به سمت خروجی به صورت پیشرونده (بدون بازگشت) منتقل می‌شوند.

اصول کلیدی MLP عبارتند از:

۱. لایه‌های مخفی: بین لایه ورودی و لایه خروجی، یک یا چند لایه مخفی وجود دارد که می‌توانند ویژگی‌های غیرخطی و پیچیده‌تری از داده‌ها را استخراج کنند.
  ۲. وزن‌ها و بایاس‌ها: هر اتصال بین نورون‌ها یک وزن دارد و هر نورون یک بایاس دارد. وزن‌ها و بایاس‌ها در طول فرآیند آموزش تنظیم می‌شوند.
  ۳. تابع فعال‌سازی: هر نورون یک تابع فعال‌سازی دارد که می‌تواند غیرخطی باشد، مانند تابع سیگموئید، تانژانت هایپربولیک یا ReLU. این توابع به شبکه اجازه می‌دهند تا الگوهای غیرخطی را یاد بگیرند.
  ۴. پیش‌خوراندگی و پس‌خوراندگی: در فرآیند آموزش، داده‌ها ابتدا به صورت پیشرونده از ورودی تا خروجی انتقال می‌یابند (پیش‌خوراندگی)، سپس خطای بین خروجی پیش‌بینی شده و خروجی واقعی محاسبه می‌شود و از طریق شبکه به عقب منتقل می‌شود (پس‌خوراندگی) تا وزن‌ها به روزرسانی شوند.
  ۵. بهینه‌سازی: تکنیک‌های بهینه‌سازی مانند (SGD گرادیان نزولی تصادفی، Adam یا RMSprop) برای به روزرسانی وزن‌ها و کمینه کردن یک تابع هزینه استفاده می‌شوند.
- MLP‌ها می‌توانند برای انواع مختلفی از مسائل، از جمله طبقه‌بندی، رگرسیون، و تقریب تابع استفاده شوند. با وجودی که قدرتمند هستند، MLP‌ها مستعد اورفیتینگ هستند، به خصوص وقتی که تعداد نورون‌ها یا لایه‌های مخفی زیاد باشد. برای مقابله با این مسئله، تکنیک‌هایی مانند نرمال‌سازی و قطع اتصال (dropout) استفاده می‌شود. regularization)

## Ensemble Learning - ۴-۶

متد انسembل یادگیری ماسین به روش‌هایی گفته می‌شود که از ترکیب چندین مدل پیش‌بینی‌کننده برای بهبود دقت پیش‌بینی‌ها و کاهش واریانس استفاده می‌کنند. این روش‌ها می‌توانند دقت کلی یک سیستم را از طریق ترکیب تصمیمات چندین مدل افزایش دهند، که هر یک به تنها ی ممکن است نقاط ضعف خاصی داشته باشند.

ما یک نمایش بصری از یک رده‌بند انتخاب‌گری (Voting Classifier) در کتابخانه‌ی `scikit-learn` در کتابخانه‌ی

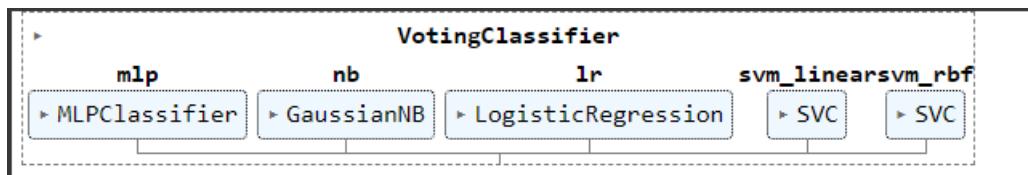
نشان داده ایم. این رده‌بند انتخاب‌گری از چندین رده‌بند مختلف استفاده می‌کند:

۱. **MLPClassifier**: یک شبکه عصبی چند لایه که قادر به یادگیری الگوهای پیچیده است.

۲. **GaussianNB**: یک رده‌بند نایو بیز که از توزیع‌های نرمال برای پیش‌بینی استفاده می‌کند.

۳. **Logistic Regression**: یک رگرسیون لجستیک برای پیش‌بینی احتمالات رویدادهای بایزی.

۴. **SVM**: ماشین بردار پشتیبانی با استفاده ازتابع هسته، که می‌تواند خطی (LinearSVM) یا شعاعی (RBF) باشد.



در یک رده‌بند انتخاب‌گری، هر یک از این مدل‌ها به طور مستقل پیش‌بینی خود را ارائه می‌دهند، و سپس این پیش‌بینی‌ها بر اساس رأی گیری اکثریت یا میانگین‌گیری وزن‌دار (بسته به تنظیم رده‌بند) ترکیب می‌شوند تا پیش‌بینی نهایی به دست آید. این روش انسembل می‌تواند به خصوص در مواردی که مدل‌های فردی به تنها ی ممکن است دارای بایاس یا واریانس بالایی باشند، بسیار موثر باشد. از آنجا که هر مدل ممکن است به نواحی مختلفی از فضای ویژگی حساس باشد، ترکیب آن‌ها می‌تواند منجر به مدل نهایی شود که از نقاط قوت هر یک بهره می‌برد و نقاط ضعف را پوشش می‌دهد.

این تکنیک انسembل به خصوص در مسائل پیچیده‌تر که هیچ مدل فردی به تنها ی نمی‌تواند الگوهای پنهان در داده‌ها را کاملاً بیابد، موثر است. انتخاب ترکیب مناسبی از رده‌بندها و تنظیم آن‌ها به شیوه‌ای که با یکدیگر خوب کار کنند، کلید موفقیت در استفاده از رده‌بند انتخاب‌گری است.

## ۲-۵-نرمال‌سازی

از ماثول `StandardScaler` استفاده برای استانداردسازی ویژگی‌ها می‌شود. این فرآیند باعث می‌شود که هر ویژگی به گونه‌ای تغییر کند که میانگین آن صفر و واریانس آن یک شود. این کار معمولاً منجر به بهبود عملکرد الگوریتم‌های یادگیری ماشین می‌شود، زیرا بسیاری از مدل‌ها (به خصوص آن‌هایی که از روش‌های گرادیان نزولی برای یادگیری استفاده می‌کنند) بهتر با داده‌هایی کار می‌کنند که همه ویژگی‌ها در مقیاس یکسانی قرار دارند. از قطعه کد زیر برای نرمال‌سازی استفاده نمودیم.

```
# Initialize the StandardScaler
scaler = StandardScaler()

# Fit and transform the scaler on the training data
X_train = scaler.fit_transform(X_train)

# Transform the testing data using the same scaler
X_test = scaler.transform(X_test)
```

- این متدهای آموزشی اجرا می‌شود تا پارامترهای مورد نیاز برای تبدیل (یعنی میانگین و انحراف معیار هر ویژگی) را یاد بگیرد و سپس تبدیل را روی داده‌های آموزشی انجام دهد.

- این متدهای آزمایشی اجرا می‌شود تا با استفاده از همان پارامترهای یاد گرفته شده از داده‌های آموزشی، داده‌های آزمایشی را تبدیل کند.

استفاده از همان پارامترها برای هر دو تبدیل اهمیت دارد تا اطمینان حاصل شود که مدل نهایی قادر است پیش‌بینی‌های دقیق روی داده‌های جدید که همان تبدیل را تجربه نکرده‌اند، انجام دهد. این امر همچنین از نشت داده‌ها (data leakage) جلوگیری می‌کند، که می‌تواند زمانی رخ دهد که اطلاعاتی از مجموعه داده‌های آزمایشی به طور ناخواسته در فرآیند آموزش استفاده شود.

نرمال‌سازی یک مرحله مهم در پیش‌پردازش داده‌های یادگیری ماشین است زیرا الگوریتم‌های مختلف ممکن است به شدت تحت تأثیر مقیاس و توزیع ویژگی‌ها قرار بگیرند.

## ۶-۲- کاهش ابعاد

کاهش بعد (Dimensionality Reduction) یک فرآیند مهم در پیش‌پردازش داده‌ها در یادگیری ماشین است که هدف آن کاهش تعداد ویژگی‌های موجود در داده‌ها، بدون از دست دادن اطلاعات مهم، است. این فرآیند می‌تواند به کاهش مدت زمان و حافظه مورد نیاز برای آموزش مدل‌ها، افزایش دقت، و کمک به مقابله با نفرین ابعاد (Curse of Dimensionality) کمک کند. این کار به چند دلیل مهم انجام می‌شود:

کاهش پیچیدگی محاسباتی: مدل‌هایی با ویژگی‌های کمتر، سریع‌تر آموزش می‌بینند و کمتر منابع محاسباتی مصرف می‌کنند.

کاهش overfitting: وقتی تعداد ویژگی‌ها نسبت به تعداد نمونه‌ها زیاد باشد، مدل ممکن است نویزهای موجود در داده‌های آموزشی را یاد بگیرد که این موضوع می‌تواند به overfitting منجر شود.

تسهیل تجسم داده‌ها: کاهش بعدیت به ما اجازه می‌دهد تا داده‌هایی با ابعاد بالا را در دو یا سه بعد تصویر کنیم، که این موضوع درک بهتری از ساختار داده‌ها و الگوهای موجود در آن‌ها را فراهم می‌کند.

کاهش بعد به دو دسته کلی تقسیم می‌شود که انتخاب ویژگی و استخراج ویژگی می‌باشد.

### انتخاب ویژگی (Feature Selection)

انتخاب ویژگی شامل انتخاب یک زیرمجموعه از ویژگی‌های موجود است. این کار به سه روش اصلی انجام می‌شود:

فیلتر کردن (Filter methods): ویژگی‌ها بر اساس آماره‌هایی مانند ارتباط یا معیارهای اطلاعات متقابل انتخاب می‌شوند.

بسته‌بندی (Wrapper methods): الگوریتم‌های یادگیری ماشین برای ارزیابی ترکیب‌های مختلف ویژگی‌ها استفاده می‌شوند.

تعییه شده (Embedded methods): که در آن روش‌های انتخاب ویژگی به طور مستقیم در طراحی مدل یادگیری ماشین تعییه می‌شوند، مانند مدل‌هایی با جریمه‌های مبتنی بر نرمال‌سازی.

### استخراج ویژگی (Feature Extraction)

استخراج ویژگی به معنای تبدیل ویژگی‌های موجود به یک فضای با بعد کمتر است که هنوز هم اطلاعات اساسی موجود در داده‌ها را حفظ می‌کند. مانند PCA و LDA

## PCA-۲-۶-۱

تجزیه مؤلفه‌های اصلی (PCA) یک روش آماری است که با تبدیل اطلاعات موجود در داده‌های با ابعاد بالا به یک فضای جدید با ابعاد کمتر به نام مؤلفه‌های اصلی، سعی در حفظ بیشترین میزان واریانس دارد. این کار باعث می‌شود که اطلاعات مهم داده‌ها حفظ شوند در حالی که بعد کمتری دارند، که این می‌تواند به کارایی محاسباتی و دقت مدل‌های یادگیری ماشین کمک کند.

اینجا چگونگی کارکرد PCA را بررسی می‌کنیم:

مرکزیت داده‌ها: ابتدا داده‌ها بر اساس میانگین هر ویژگی مرکزیت می‌یابند، یعنی میانگین هر ویژگی از مقادیر آن کم می‌شود.

محاسبه کوواریانس: ماتریس کوواریانس داده‌ها ساخته می‌شود که واریانس‌ها در قطر اصلی و کوواریانس بین هر جفت ویژگی‌ها را در خارج از قطر نشان می‌دهد.

تجزیه ماتریس کوواریانس: با استفاده از تجزیه مقادیر ویژه (eigenvalue decomposition) بر روی ماتریس کوواریانس، مقادیر ویژه و بردارهای ویژه محاسبه می‌شوند.

انتخاب مؤلفه‌ها: مقادیر ویژه را می‌توان مرتب کرد تا اهمیت نسبی بردارهای ویژه که هر کدام یک مؤلفه اصلی را نشان می‌دهند، مشخص شود. سپس، می‌توان تعداد مشخصی از بردارهای ویژه که بیشترین مقادیر ویژه را دارند را انتخاب کرد، که این بردارها بیشترین واریانس را در داده‌ها توضیح می‌دهند.

تبدیل داده‌ها: در نهایت، داده‌ها به فضای جدید مؤلفه‌های اصلی تبدیل می‌شوند که توسط بردارهای ویژه انتخاب شده تعریف می‌شود. این تبدیل به صورت ضرب داده‌های مرکزیت یافته در ماتریس بردارهای ویژه انجام می‌شود.

## LDA-۲-۶-۲

تحلیل تمایزی خطی (Linear Discriminant Analysis - LDA) یک روش کاهش بعدی است که در آمار و یادگیری ماشین استفاده می‌شود، به ویژه برای مسائل طبقه‌بندی LDA بر خلاف PCA که فقط واریانس داده‌ها را در نظر می‌گیرد، سعی دارد ابعاد داده‌ها را به گونه‌ای کاهش دهد که تمایز بین کلاس‌های مختلف به حداقل برسد.

در اینجا نحوه کار LDA را بیان می‌کنیم:

۱. بین کلاس و درون کلاس **LDA**: بین واریانس داده‌ها (تفاوت بین نمونه‌های مختلف در یک کلاس) و واریانس بین کلاس‌ها (تفاوت بین میانگین‌های کلاس‌های مختلف) را در نظر می‌گیرد. هدف این است که واریانس بین کلاس‌ها را به حداقل رسانده و واریانس درون کلاس را به حداقل برسانیم.

۲. ماتریس‌های واریانس **LDA**: ماتریس واریانس درون کلاس و ماتریس واریانس بین کلاس را محاسبه می‌کند. ماتریس واریانس درون کلاس نشان‌دهنده پراکندگی داده‌ها داخل هر کلاس و ماتریس واریانس بین کلاس نشان‌دهنده پراکندگی میانگین‌های کلاس‌ها است.

۳. تابع معیار **LDA**: سپس یک تابع معیار را بهینه می‌کند که معمولاً نسبت بین واریانس‌های بین کلاس به واریانس‌های درون کلاس است. این کار با یافتن بردارهایی که به این نسبت را به حداقل می‌رسانند، انجام می‌شود.

۴. تبدیل خطی: با استفاده از بردارهای به دست آمده، داده‌ها به فضای کاهش یافته‌ای تبدیل می‌شوند که در آن، کلاس‌ها به بهترین شکل از هم جدا می‌شوند. این امر باعث می‌شود که الگوریتم‌های طبقه‌بندی مانند رگرسیون لجستیک یا ماشین‌های بردار پشتیبانی بتوانند به طور موثرتری کار کنند.

۵. کاربردها **LDA**: علاوه بر کاهش ابعاد، می‌تواند به عنوان یک الگوریتم طبقه‌بندی نیز عمل کند. به عنوان یک روش کاهش بعدی، اغلب قبل از اجرای طبقه‌بندی استفاده می‌شود تا ویژگی‌های بیشتر مرتبط با تفاوت‌های کلاس‌ها حفظ شوند.

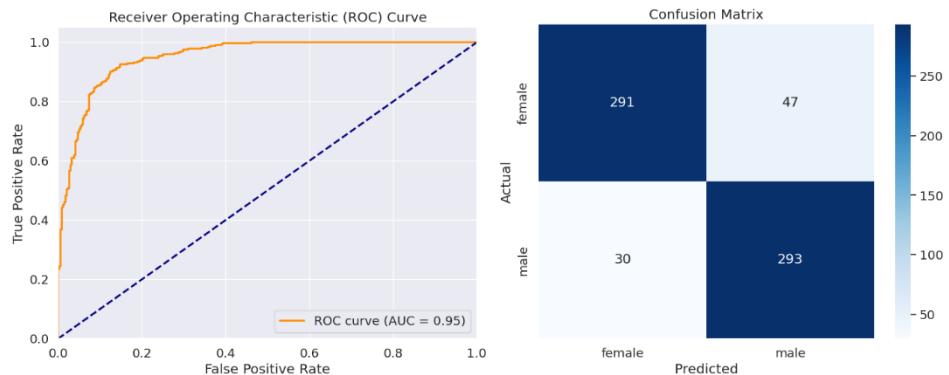
LDA به ویژه زمانی مفید است که تعداد نمونه‌ها نسبت به تعداد ویژگی‌ها کم است، چرا که با کاهش بعدی داده‌ها، می‌تواند از پدیده نفرین بعدیت (Curse of Dimensionality) جلوگیری کند و اورفیتینگ را کاهش دهد.

## ۲-۷-بررسی نتایج اجرای طبقه‌بندهای معرفی شده

در این قسمت به توضیح و تفسیر نتایج اجرای طبقه‌بندهای مختلف خواهیم پرداخت. در ابتدا بدون کاهش بعد و با نرمال‌سازی طبقه‌بندی را انجام خواهیم داد.

### ۲-۷-۱ **Linear SVM**-بدون کاهش بعد و با نرمال‌سازی

این مدل بسیار ساده بوده و تنها قابلیت جداسازی داده‌ها به صورت خطی را دارا می‌باشد. با این حال به نتایج نسبتاً مناسبی بر روی داده‌های ما رسیده است.



شکل ۶. ماتریس درهمریختگی و منحنی ROC بدون کاهش بعد و با نرمال‌سازی

با توجه به شکل ۶ که حاوی ماتریس درهمریختگی می‌باشد، متوجه عملکرد خوب مدل می‌شویم. همچنین مساحت زیر نمودار AUC برابر با ۰,۹۵ می‌باشد و مدل در هر دو کلاس مرد و زن تقریباً به یک شکل عمل کرده است. همچنین نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۷ قابل مشاهده هستند.

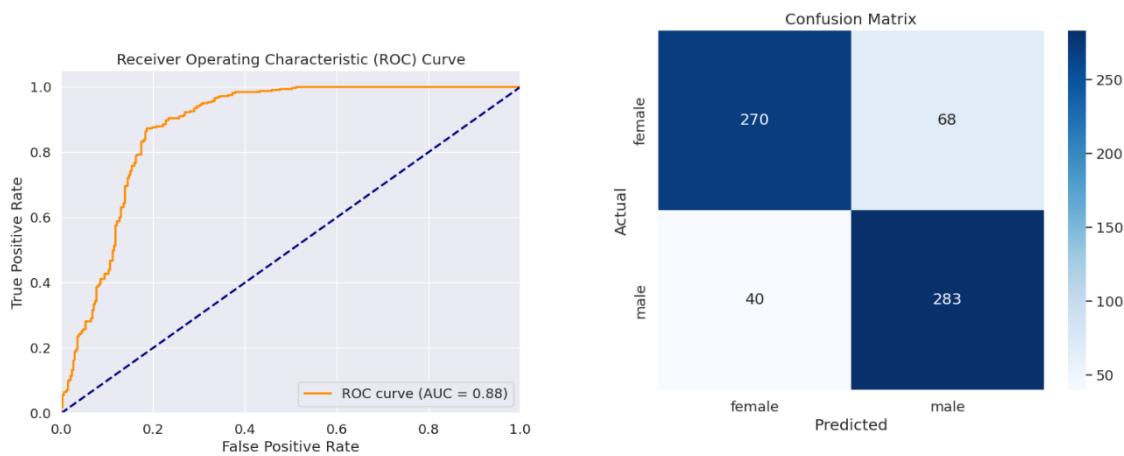
Classification Report:				
	precision	recall	f1-score	support
0	0.91	0.86	0.88	338
1	0.86	0.91	0.88	323
accuracy			0.88	661
macro avg	0.88	0.88	0.88	661
weighted avg	0.88	0.88	0.88	661

Precision: 0.861764705882353  
 Recall: 0.9071207430340558  
 F1 score: 0.8838612368024134

شکل ۷. نتایج ارزیابی کلاسی و کلی داده برای Linear SVM بدون کاهش بعد و با نرمال‌سازی

#### Naïve Bayes-۲-۷-۲

این مدل نیز بسیار ساده بوده و با فرض مرتبط نبودن ویژگی‌ها و قانون بیز تلاش بر دسته‌بندی داده‌ها دارد، با این حال این مدل نیز عملکرد نسبتاً مناسبی برای هر دو کلاس داشته است که ماتریس درهمریختگی و منحنی ROC در شکل ۸ قابل مشاهده است.



شکل ۸. ماتریس درهمریختگی و منحنی ROC برای Naive Bayes بدون کاهش بعد و با نرماسازی

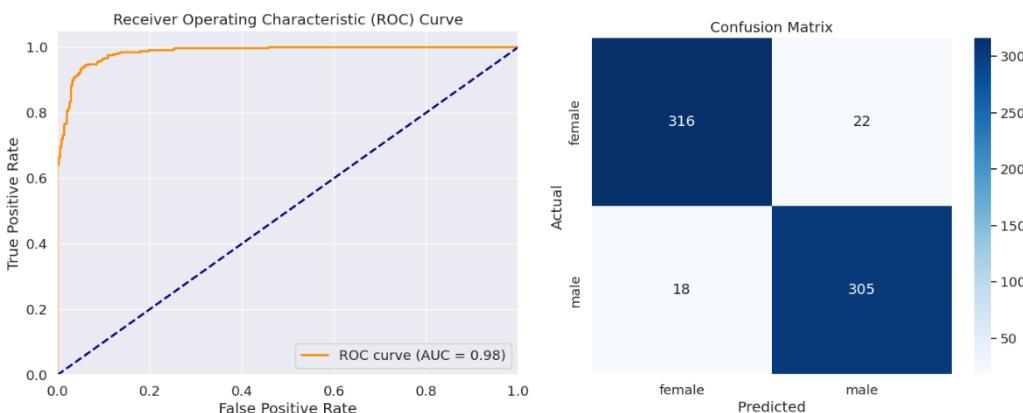
با توجه به ماتریس درهمریختگی متوجه عملکرد خوب و مناسب مدل می‌شویم. همچنین مساحت زیر نمودار AUC برابر با  $0.88$  می‌باشد و مدل در هر دو کلاس مرد و زن تقریباً به یک شکل عمل کرده است با این حال این مدل از مدل SVM بدتر عمل کرده است. همچنین نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۹ قابل مشاهده هستند.

Classification Report:						
	precision	recall	f1-score	support		
Precision:	0.8062678062678063	0	0.87	0.80	0.83	338
Recall:	0.8761609907120743	1	0.81	0.88	0.84	323
F1 score:	0.8397626112759644					
	accuracy			0.84	0.84	661
	macro avg		0.84	0.84	0.84	661
	weighted avg		0.84	0.84	0.84	661

شکل ۹. نتایج ارزیابی کلاسی و کلی داده برای Naive Bayes بدون کاهش بعد و با نرماسازی

### RBF SVM-۲-۷-۳

این مدل طبق توضیحات، قابلیت دسته‌بندی ویژگی‌های پیچیده تری را دارد و توقع داریم نسبت به مدل‌های قبلی بسیار بهتر عمل کند. ماتریس درهمریختگی و منحنی ROC در شکل ۱۰ قابل مشاهده است.



شکل ۱۰. ماتریس درهمریختگی و منحنی ROC برای RBF SVM بدون کاهش بعد و با نرم‌السازی

با توجه به ماتریس درهمریختگی شکل ۱۰، عملکرد بسیار خوب مدل مشخص می‌شود. همچنین مساحت زیر نمودار AUC برابر با ۰,۹۸ می‌باشد و مدل در هر دو کلاس مرد و زن تقریباً به یک شکل عمل کرده است. همچنین نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۱۱ قابل مشاهده هستند.

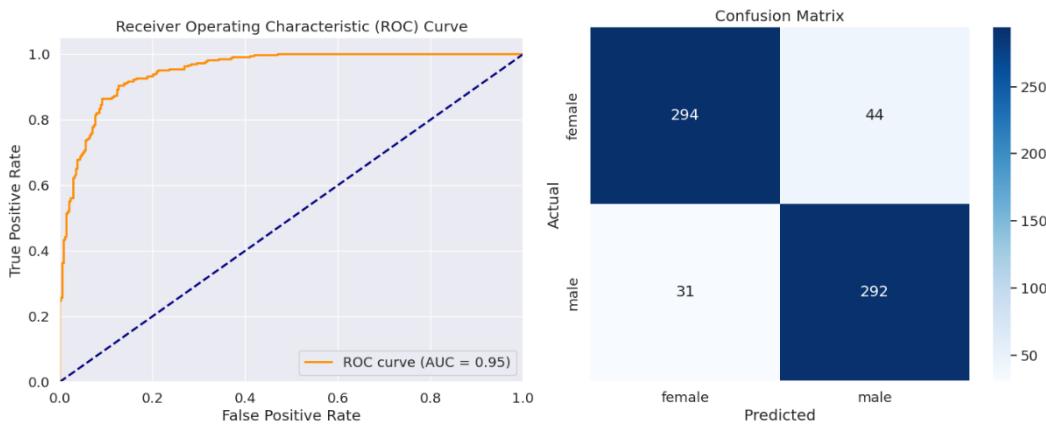
Classification Report:					
	precision	recall	f1-score	support	
0	0.95	0.93	0.94	338	
1	0.93	0.94	0.94	323	
Precision:	0.9327217125382263				
Recall:	0.9442724458204335	accuracy	0.94	661	
F1 score:	0.9384615384615386	macro avg	0.94	0.94	661
		weighted avg	0.94	0.94	661

شکل ۱۱. نتایج ارزیابی کلاسی و کلی داده برای RBF SVM بدون کاهش بعد و با نرم‌السازی

این مدل از دو مدل قبلی به شدت بهتر عمل کرده است و دلیل آن مشخصاً به خاطر خاصیت غیر خطی می‌باشد.

#### ۲-۷-۴ Logistic Regression

این مدل نیز بسیار ساده بوده و فقط قابلیت جداسازی داده‌ها به صورت خطی را دارا می‌باشد، با این حال نتایج نسبتاً مناسبی بر روی داده‌های ما دریافت می‌کند.



شکل ۱۲. ماتریس درهمیریختگی و منحنی ROC برای Logistic Regression بدون کاهش بعد و با نرمال‌سازی

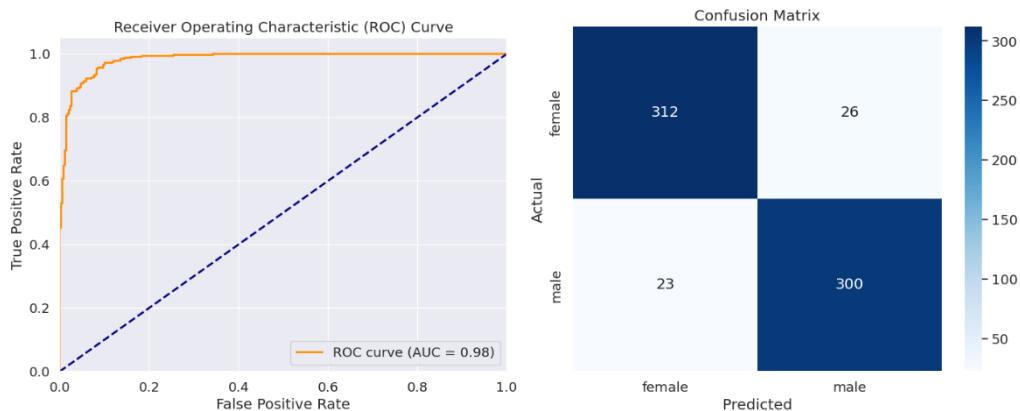
با توجه به ماتریس درهمیریختگی شکل ۱۲، به عملکرد خوب مدل نشان داده می‌شود. همچنین مساحت زیر نمودار AUC برابر با  $0,94$  می‌باشد و مدل در هر دو کلاس مرد و زن تقریباً به یک شکل عمل کرده است. همچنین نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۱۳ قابل مشاهده هستند.

Classification Report:					
	precision	recall	f1-score	support	
Precision: 0.8690476190476191	0	0.90	0.87	0.89	338
Recall: 0.9040247678018576	1	0.87	0.90	0.89	323
F1 score: 0.8861911987860394			accuracy	0.89	661
			macro avg	0.89	661
			weighted avg	0.89	661

شکل ۱۳. نتایج ارزیابی کلاسی و کلی داده برای Logistic Regression بدون کاهش بعد و با نرمال‌سازی

#### MLP-۲-۷-۵ بدون کاهش بعد و با نرمال‌سازی

شبکه‌ی عصبی چند لایه قابلیت، دسته‌بندی داده‌ها به شدت پیچیده را دارد، لذا ما توقع داریم این مدل بهترین عملکرد را نسبت به سایر مدل‌ها به ما بدهد.



شکل ۱۴. ماتریس درهمریختگی و منحنی ROC برای MLP بدون کاهش بعد و با نرمال‌سازی

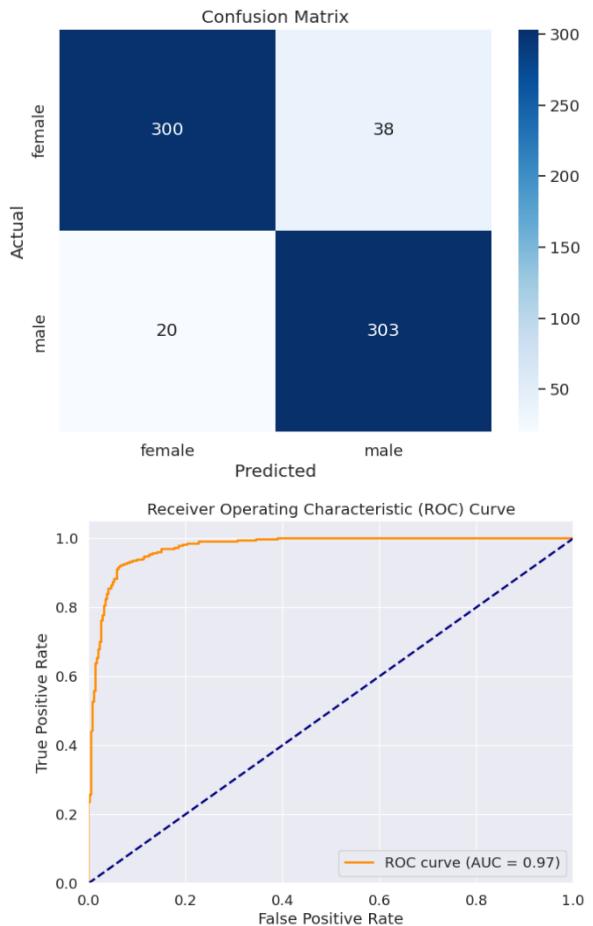
با توجه به ماتریس درهمریختگی شکل ۱۴، عملکرد بسیار خوب مدل را ملاحظه می‌نماییم. همچنین مساحت زیر نمودار AUC برابر با  $0.98$  می‌باشد و مدل در هر دو کلاس مرد و زن تقریباً به یک شکل عمل کرده است. همچنین نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۱۵ قابل مشاهده هستند.

Classification Report:					
	precision	recall	f1-score	support	
Precision: 0.9202453987730062	0	0.93	0.92	0.93	338
Recall: 0.9287925696594427	1	0.92	0.93	0.92	323
F1 score: 0.9244992295839752			accuracy	0.93	661
			macro avg	0.93	0.93
			weighted avg	0.93	0.93

شکل ۱۵. نتایج ارزیابی کلاسی و کلی داده برای MLP بدون کاهش بعد و با نرمال‌سازی

## ۶-۷-۲- Ensemble Method

با توجه به اینکه تمام مدل‌ها معرفی شده نسبتاً عملکرد مناسبی برای این داده‌ها داشتن، توقع می‌رود این مدل نیز عملکرد خوبی داشته باشد و البته روی داده‌ی تست عمومیت پذیری بیشتری داشته باشد.



شکل ۱۶. ماتریس درهمبرختگی و منحنی ROC برای Ensemble models بدون کاهش بعد و با نرمال‌سازی

با توجه به ماتریس درهمبرختگی شکل ۱۶، متوجه به عملکرد خوب مدل می‌شویم. همچنین مساحت زیر نمودار AUC برابر با  $0,97$  می‌باشد و مدل در هر دو کلاس مرد و زن تقریباً به یک شکل عمل کرده است. همچنین نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۱۷ قابل مشاهده هستند.

Classification Report:					
	precision	recall	f1-score	support	
0	0.94	0.89	0.91	338	
1	0.89	0.94	0.91	323	
accuracy			0.91	661	
macro avg	0.91	0.91	0.91	661	
weighted avg	0.91	0.91	0.91	661	

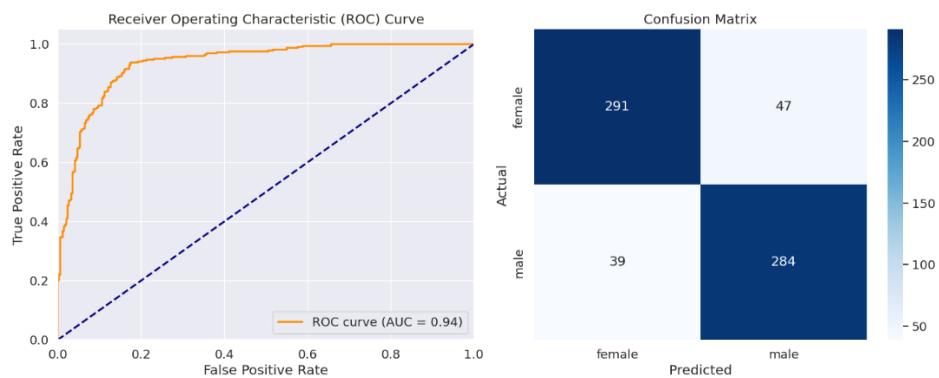
Precision: 0.8885630498533724  
Recall: 0.9380804953560371  
F1 score: 0.9126506024096386

شکل ۱۷. نتایج ارزیابی کلاسی و کلی داده برای Ensemble Models بدون کاهش بعد و با نرمال‌سازی

حال بدون نرمال‌سازی مجدداً سراغ بررسی مدل‌های معرفی شده خواهیم رفت.

## Linear SVM-۲-۷-۷ بدون کاهش بعد و بدون نرمال سازی

این مدل بسیار ساده بوده و فقط قابلیت جداسازی داده ها به صورت خطی را دارد می باشد، با این حال نتایج نسبتاً مناسبی بر روی داده های ما دریافت می کند.



شکل ۱۸. ماتریس درهم ریختگی و منحنی ROC برای Linear SVM بدون کاهش بعد و بدون نرمال سازی

با توجه به ماتریس درهم ریختگی شکل ۱۸ متوجه به عملکرد خوب مدل می شویم، همچنین مساحت زیر نمودار AUC برابر با  $0.94$  می باشد و مدل در هر دو کلاس مرد و زن تقریباً به یک شکل عمل کرده است.

ماتریس درهم ریختگی نشان دهنده تعداد پیش بینی های صحیح و نادرست مدل است. اعداد روی محور عمودی (Actual) نشان دهنده کلاس های واقعی هستند و اعداد روی محور افقی (Predicted) نشان دهنده پیش بینی های مدل هستند. برای مثال، مدل ۲۹۱ بار به درستی کلاس 'female' را پیش بینی کرده و ۴۷ بار اشتباهآ کلاس 'male' را به عنوان 'female' پیش بینی کرده است. به طور مشابه، برای کلاس 'male' مدل ۲۸۴ بار به درستی و ۳۹ بار اشتباهآ پیش بینی کرده است.

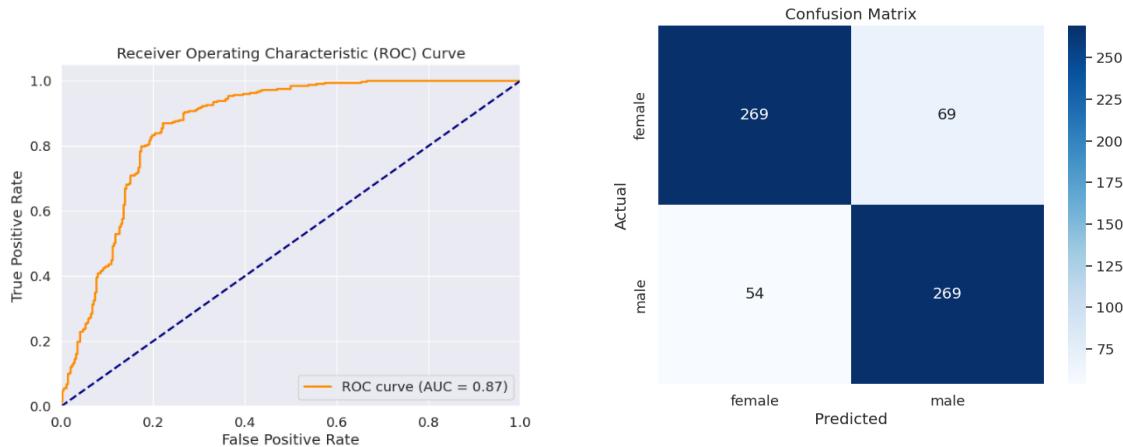
همچنین نتایج ارزیابی داده های تست مربوط به هر کلاس و به صورت کلی در شکل ۱۹ قابل مشاهده هستند.

Classification Report:				
	precision	recall	f1-score	support
0	0.88	0.86	0.87	338
1	0.86	0.88	0.87	323
accuracy			0.87	661
macro avg	0.87	0.87	0.87	661
weighted avg	0.87	0.87	0.87	661

Precision: 0.8580060422960725  
 Recall: 0.8792569659442725  
 F1 score: 0.8685015290519877

شکل ۱۹. نتایج ارزیابی کلاسی و کلی داده برای Linear SVM بدون کاهش بعد و بدون نرمال سازی

## Naïve Bayes-۲-۷-۸ بدون کاهش بعد و بدون نرمال سازی



شکل ۲۰. ماتریس درهمیریختگی و منحنی ROC برای Naive Bayes بدون کاهش بعد و بدون نرمال سازی

با توجه به ماتریس درهمیریختگی شکل ۲۰، متوجه به عملکرد خوب مدل می‌شویم، همچنین مساحت زیر نمودار AUC برابر با  $0.87$  می‌باشد و مدل در هر دو کلاس زن تقریبا بدتر عمل کرده است با این حال این مدل از مدل SVM بدتر عمل کرده است. همچنین نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۲۱ قابل مشاهده هستند.

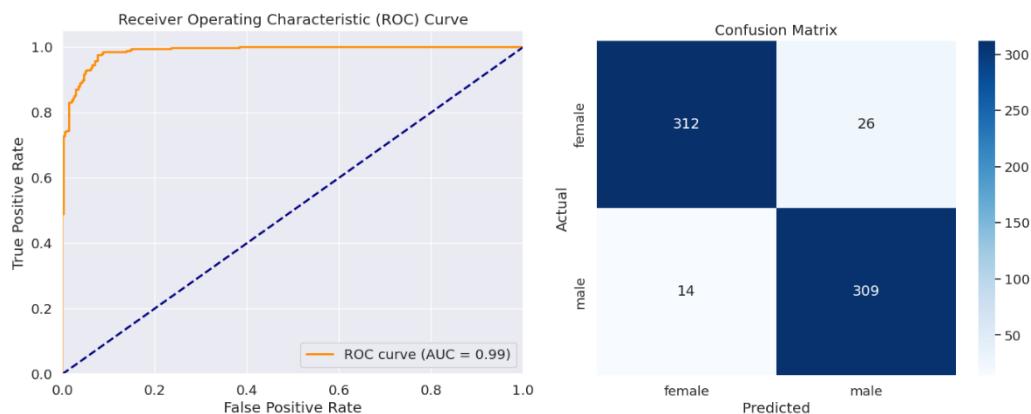
در مجموع، این نتایج نشان می‌دهند که مدل Naive Bayes بدون نرمال سازی داده‌ها دارای دقت کمتری نسبت به مدل SVM خطی قبلی است. نرمال سازی داده‌ها می‌تواند به بهبود دقت مدل کمک کند، زیرا تاثیر ویژگی‌های با مقیاس‌های مختلف را کاهش می‌دهد و به مدل اجازه می‌دهد تا ویژگی‌ها را به طور موثرتری یاد بگیرد.



شکل ۲۱. نتایج ارزیابی کلاسی و کلی داده برای Naive Bayes بدون کاهش بعد و بدون نرمال سازی

## RBF SVM-۲-۷-۹ بدون کاهش بعد و بدون نرمالسازی

این مدل قابلیت دسته‌بندی ویژگی‌های پیچیده تری را دارد و توقع داریم نسبت به مدل‌های قبلی بسیار بهتر عمل کند. ماتریس درهمریختگی و منحنی ROC در شکل ۲۲ قابل مشاهده است.



شکل ۲۲. ماتریس درهمریختگی و منحنی ROC برای RBF SVM بدون کاهش بعد و بدون نرمالسازی

با توجه به ماتریس درهمریختگی متوجه به عملکرد بسیار خوب مدل می‌شویم، همچنین مساحت زیر نمودار AUC برابر با ۰,۹۹ می‌باشد و مدل در هر کلاس زن تقریباً بهتر عمل کرده است.

در مجموع، این نتایج نشان‌دهنده‌ی عملکرد فوق‌العاده‌ی مدل SVM با هسته‌ی RBF است، حتی بدون نرمالسازی داده‌ها. با این حال، باید توجه داشت که نرمالسازی داده‌ها عموماً توصیه می‌شود زیرا می‌تواند به کاهش تأثیر ویژگی‌های با مقیاس‌های بسیار متفاوت کمک کند و به این ترتیب ممکن است به بهبود بیشتر عملکرد کمک کند، خصوصاً در شرایطی که داده‌های جدید ممکن است متفاوت از داده‌های آموزشی باشند.

همچنین نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۲۳ قابل مشاهده هستند.

Classification Report:					
	precision	recall	f1-score	support	
0	0.96	0.92	0.94	338	
1	0.92	0.96	0.94	323	
accuracy			0.94	661	
macro avg	0.94	0.94	0.94	661	
weighted avg	0.94	0.94	0.94	661	

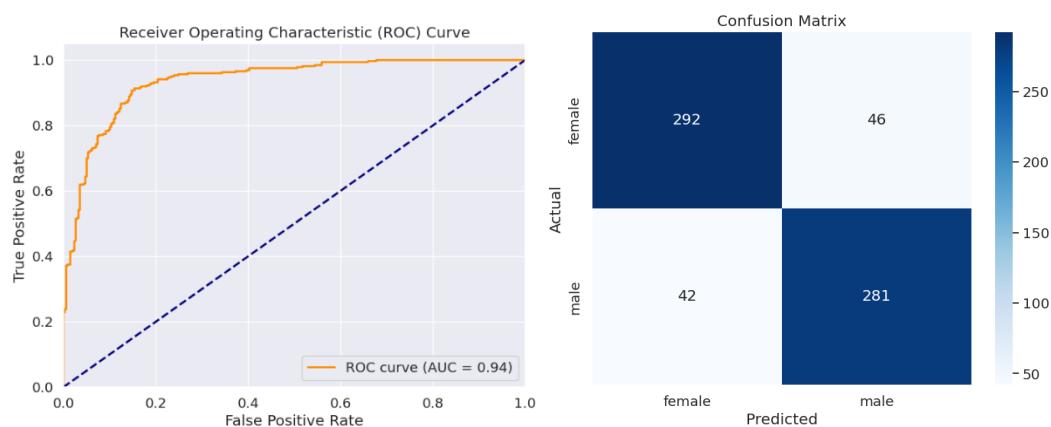
Precision: 0.9223880597014925  
 Recall: 0.9566563467492261  
 F1 score: 0.939209726443769

شکل ۲۳. نتایج ارزیابی کلاسی و کلی داده برای RBF SVM بدون کاهش بعد و بدون نرمالسازی

این مدل از دو مدل قبلی به شدت بهتر عمل کرده است و دلیل آن مشخصاً به خاطر خاصیت غیر خطی می‌باشد.

#### ۲-۷-۱۰ Logistic Regression- بدون کاهش بعد و بدون نرمال‌سازی

این مدل بسیار ساده بوده و فقط قابلیت جداسازی داده‌ها به صورت خطی را دارد. با این حال نتایج نسبتاً مناسبی بر روی داده‌های ما دریافت می‌کند.



شکل ۲۴. ماتریس درهمریختگی و منحنی ROC برای Logistic Regression بدون کاهش بعد و بدون نرمال‌سازی

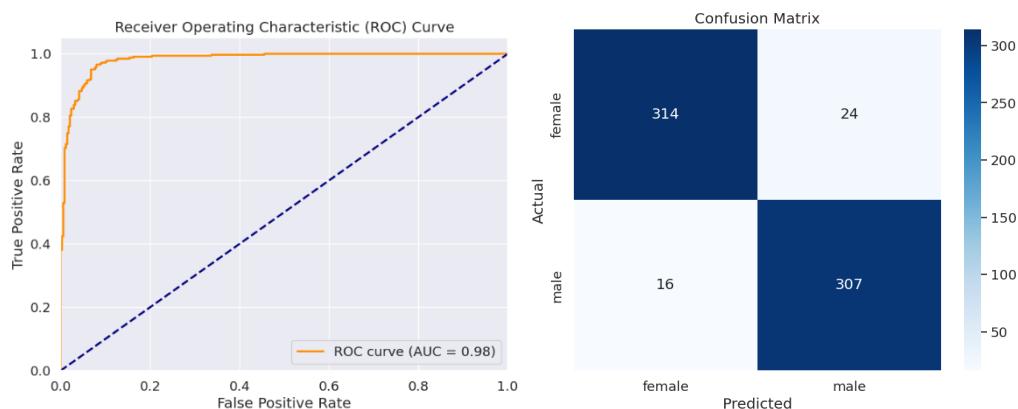
با توجه به ماتریس درهمریختگی شکل ۲۴، متوجه به عملکرد خوب مدل می‌شویم، همچنین منحنی ROC نمایانگر یک عملکرد خوب با AUC (مساحت زیر منحنی) برابر با ۰,۹۴ است. این مقدار نشان‌دهنده توانایی خوب مدل در تمایز دادن بین دو کلاس است، منحنی نسبتاً نزدیک به گوشه بالا سمت چپ قرار دارد، که نشان‌دهنده نرخ بالای مثبت واقعی و نرخ پایین مثبت کاذب است. همچنین نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۲۵ قابل مشاهده هستند.

Classification Report:					
	precision	recall	f1-score	support	
Precision: 0.8593272171253823	0	0.87	0.86	0.87	338
Recall: 0.8699690402476781	1	0.86	0.87	0.86	323
F1 score: 0.8646153846153847					
	accuracy			0.87	661
	macro avg	0.87	0.87	0.87	661
	weighted avg	0.87	0.87	0.87	661

شکل ۲۵. نتایج ارزیابی کلاسی و کلی داده برای Logistic Regression بدون کاهش بعد و بدون نرمال‌سازی

## MLP-۲-۷-۱۱ بدون کاهش بعد و بدون نرمال سازی

شبکه‌ی عصبی چند لایه قابلیت، دسته‌بندی داده‌ها به شدت پیچیده را دارد، لذا ما توقع داریم این مدل بهترین عملکرد را نسبت به سایر مدل‌ها به ما بدهد.



شکل ۲۶. ماتریس درهمربختگی و منحنی ROC برای MLP بدون کاهش بعد و بدون نرمال سازی

با توجه به ماتریس درهمربختگی شکل ۲۶، متوجه به عملکرد خوب مدل می‌شویم، همچنین منحنی ROC نشان‌دهنده عملکرد عالی مدل است با AUC (مساحت زیر منحنی) برابر با ۰،۹۸. این مقدار نشان می‌دهد که مدل توانایی فوق العاده‌ای در تمایز دادن بین دو کلاس 'male' و 'female' دارد. منحنی خیلی نزدیک به گوشه بالا سمت چپ است، که نشان‌دهنده نرخ بالای مثبت واقعی و نرخ پایین مثبت کاذب است.

همچنین نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۲۷ قابل مشاهده هستند.

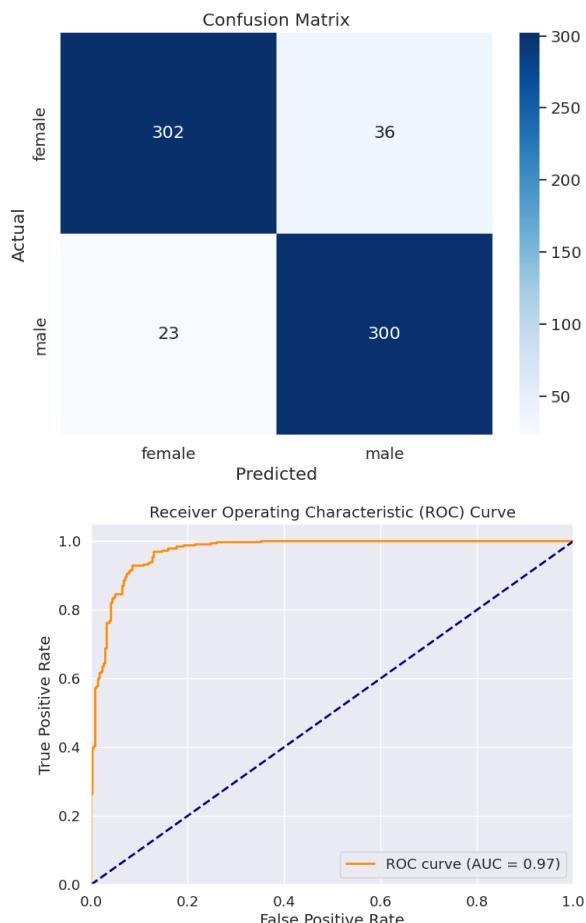
Classification Report:					
	precision	recall	f1-score	support	
0	0.95	0.93	0.94	338	
1	0.93	0.95	0.94	323	
accuracy			0.94	661	
macro avg	0.94	0.94	0.94	661	
weighted avg	0.94	0.94	0.94	661	

Precision: 0.9274924471299094  
Recall: 0.9504643962848297  
F1 score: 0.9388379204892967

شکل ۲۷. نتایج ارزیابی کلاسی و کلی داده برای MLP بدون کاهش بعد و بدون نرمال سازی

## ۱۲-۷-۲ Ensemble Method بدون کاهش بعد و بدون نرمال‌سازی

با توجه به اینکه تمام مدل‌ها معرفی شده نسبتاً عملکرد مناسبی برای این داده‌ها داشتن، توقع می‌رود این مدل نیز عملکرد خوبی داشته باشد و البته روی داده‌ی تست عمومیت پذیری بیشتری داشته باشد.



شکل ۲۸. ماتریس درهمریختگی و منحنی ROC برای Ensemble models بدون کاهش بعد و بدون نرمال‌سازی

با توجه به ماتریس درهمریختگی شکل ۲۸ متوجه به عملکرد خوب مدل می‌شویم، همچنین مساحت زیر نمودار AUC برابر با  $0.97$  می‌باشد و مدل در هر دو کلاس مرد و زن تقریباً به یک شکل عمل کرده است.

در مجموع، نتایج نشان می‌دهند که مدل یادگیری ترکیبی بدون نرمال‌سازی داده‌ها عملکرد بسیار خوبی دارد. یادگیری ترکیبی معمولاً با ترکیب چندین مدل پیش‌بینی‌کننده به منظور کاهش واریانس و افزایش دقت استفاده می‌شود، و به نظر می‌رسد که در این مورد نیز موفق بوده است.

همچنین نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۲۹ قابل مشاهده هستند.

Classification Report:				
	precision	recall	f1-score	support
0	0.93	0.89	0.91	338
1	0.89	0.93	0.91	323
accuracy			0.91	661
macro avg	0.91	0.91	0.91	661
weighted avg	0.91	0.91	0.91	661

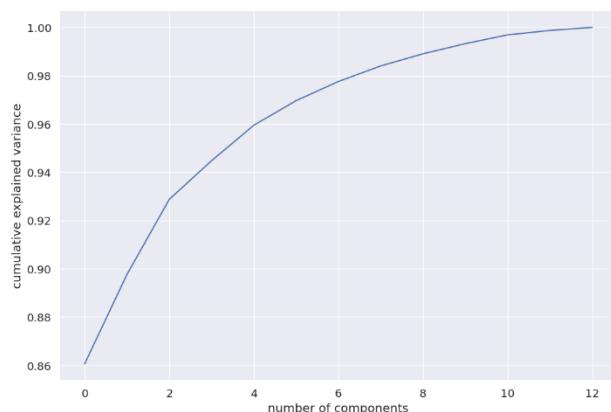
  

Precision: 0.8928571428571429
Recall: 0.9287925696594427
F1 score: 0.9104704097116844

شکل ۲۹. تایج ارزیابی کلاسی و کلی داده برای Ensemble Models بدون کاهش بعد و بدون نرمال‌سازی

## ۲-۸-کاهش بعد با استفاده از PCA

با استفاده از PCA می‌توانیم ابعاد داده‌ها را کاهش دهیم، به نحوی که داده‌ها بیشترین پراکندگی را حفظ کنند، برای این کار ابتدا بررسی می‌کنیم هر تعداد از ویژگی‌ها چقدر پراکندگی داده اصلی را توجیه می‌کنند.



شکل ۳۰. توجیه پراکندگی تجمعی با PCA

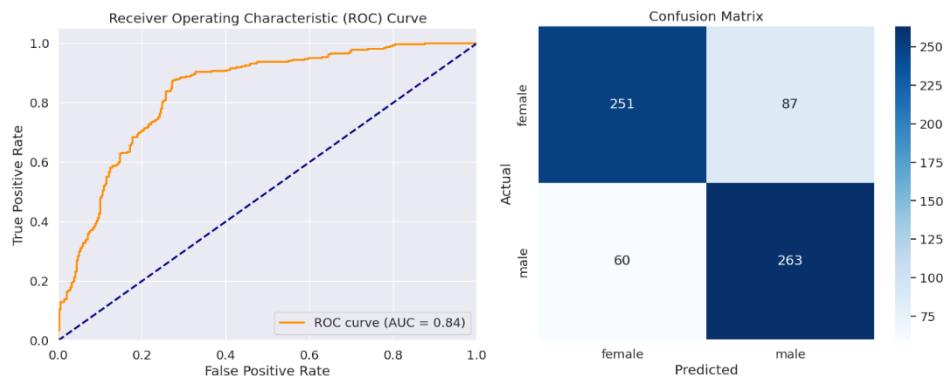
با توجه به شکل ۳۰، با ۵ ویژگی تقریبا ۹۵ درصد از پراکندگی داده توجیه می‌شود بنابراین ما ۵ ویژگی را برای استفاده انتخاب می‌کنیم، همچنان در صورت استفاده از دو ویژگی پراکندگی داده‌ها در دو کلاس به صورت شکل ۳۱ می‌باشد:



شکل ۳۱. پراکندگی داده‌ها با دو ویژگی در PCA

حال با داده‌های نرمال‌سازی شده، سراغ آموزش مجدد مدل‌ها با کاهش بعد خواهیم رفت.

#### ۲-۸-۱ با کاهش ابعاد و با نرمال‌سازی Linear SVM



شکل ۳۲. ماتریس درهمریختگی و منحنی ROC برای Linear SVM با کاهش بعد PCA و با نرمال‌سازی

با توجه به ماتریس درهمریختگی شکل ۳۲، متوجه به عملکرد خوب مدل می‌شویم، همچنین مساحت زیر نمودار AUC برابر با  $0.84$  می‌باشد.

ماتریس اشتباه: در این ماتریس، مدل در تشخیص کلاس 'female' ۲۵۱ بار به درستی و ۸۷ بار به اشتباه پیش‌بینی کرده است، و در تشخیص کلاس "male" ۲۶۳ بار به درستی و ۶۰ بار به اشتباه پیش‌بینی کرده است. این نشان‌دهنده تعداد خطاهای بیشتری نسبت به نمونه‌های قبلی است، به خصوص در پیش‌بینی کلاس 'female'.

استفاده از PCA به معنای کاهش بُعد داده‌ها و حفظ ویژگی‌هایی است که بیشترین واریانس را دارند، که می‌تواند به مدل کمک کند تا بهتر تمرکز کند و به حذف ویژگی‌های کم اهمیت کمک کند. با این حال،

در این مورد خاص، به نظر می‌رسد که ترکیب PCA و SVM خطی با داده‌های نرمال‌سازی شده به خوبی سایر مدل‌ها عمل نکرده است. این ممکن است به دلیل از دست دادن اطلاعات مهم هنگام کاهش بُعد باشد، یا شاید ویژگی‌هایی که حذف شده‌اند برای تشخیص کلاس‌ها مهم بوده‌اند. همچنین نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۳۲ قابل مشاهده هستند.

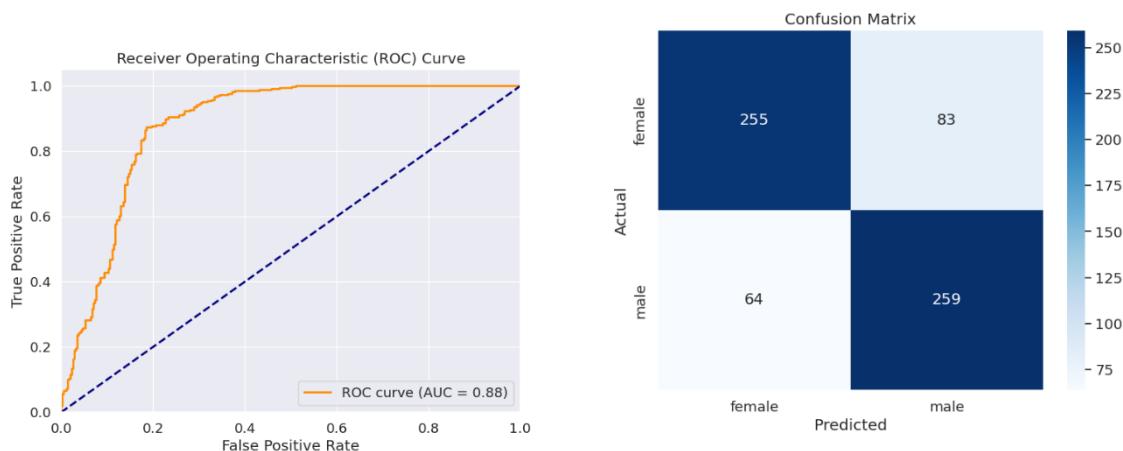
Classification Report:				
	precision	recall	f1-score	support
0	0.81	0.74	0.77	338
1	0.75	0.81	0.78	323
accuracy			0.78	661
macro avg	0.78	0.78	0.78	661
weighted avg	0.78	0.78	0.78	661

Precision: 0.7514285714285714  
 Recall: 0.8142414860681114  
 F1 score: 0.7815750371471025

شکل ۳۳. نتایج ارزیابی کلاسی و کلی داده برای Linear SVM با کاهش بعد PCA و با نرمال‌سازی

## ۲-۸-۲ Naïve Bayes با کاهش ابعاد و با نرمال‌سازی

این مدل نیز بسیار ساده بوده و با فرض مرتبط نبودن ویژگی‌ها و قانون بیز تلاش بر دسته‌بندی داده‌ها دارد، با این حال این مدل نیز عملکرد نسبتاً مناسبی برای هر دو کلاس داشته است که ماتریس درهمریختگی و منحنی ROC در شکل ۳۴ قابل مشاهده است.



شکل ۳۴. ماتریس درهمریختگی و منحنی ROC برای Naive Bayes با کاهش بعد PCA و با نرمال‌سازی

منحنی ROC دارای AUC (مساحت زیر منحنی) ۰,۸۸ است، که نشان‌دهنده عملکرد خوبی است، اما نه به خوبی برخی از مدل‌های دیگر که بررسی شده‌اند AUC کمتر از ۰,۹ نشان می‌دهد که مدل ممکن است در برخی موارد با تشخیص دقیق دو کلاس دشواری داشته باشد.

ماتریس اشتباه: ماتریس اشتباه نشان می‌دهد که مدل در تشخیص کلاس 'female' 255 بار به درستی و ۸۳ بار به اشتباه پیش‌بینی کرده است، و برای کلاس 'male' ۲۵۹ بار به درستی و ۶۴ بار به اشتباه پیش‌بینی کرده است. این نشان‌دهنده تعداد خطاهای بیشتری در مقایسه با مدل‌هایی مانند MLP یا یادگیری ترکیبی است.

استفاده از PCA می‌تواند به کاهش ابعاد داده‌ها و تمرکز بر ویژگی‌های اصلی کمک کند، و نرمال‌سازی داده‌ها می‌تواند به استانداردسازی مقیاس‌های ویژگی‌ها کمک کند. با این حال، در این مورد خاص، به نظر می‌رسد که مدل PCA نتوانسته به خوبی عملکرد مدل‌های قبلی را تکرار کند. این ممکن است به دلیل خصوصیات خاص الگوریتم Naive Bayes باشد که فرض استقلال شرطی بین ویژگی‌ها را دارد، که ممکن است در داده‌های کاهش‌یافته توسط PCA دیگر صادق نباشد. همچنین نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۳۵ قابل مشاهده هستند.

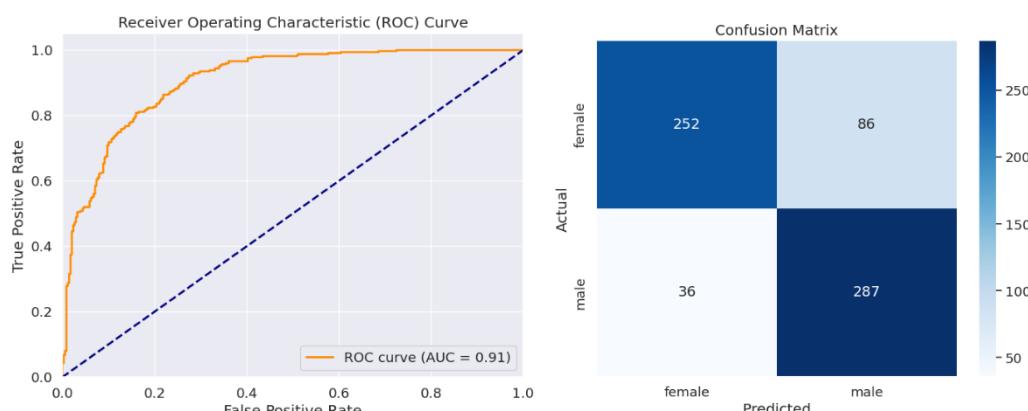
Classification Report:					
	precision	recall	f1-score	support	
0	0.85	0.76	0.80	338	
1	0.77	0.85	0.81	323	
	accuracy		0.81	661	
	macro avg		0.81	0.81	661
	weighted avg		0.81	0.81	661

Precision: 0.773109243697479  
Recall: 0.8544891640866873  
F1 score: 0.811764705882353

شکل ۳۵. نتایج ارزیابی کلاسی و کلی داده برای Naive Bayes با کاهش بعد PCA و با نرمال‌سازی

### RBF SVM-۲-۸-۳ با کاهش ابعاد و با نرمال‌سازی



شکل ۳۶. ماتریس درهمریختگی و منحنی ROC برای RBF SVM با کاهش بعد PCA و با نرمال‌سازی

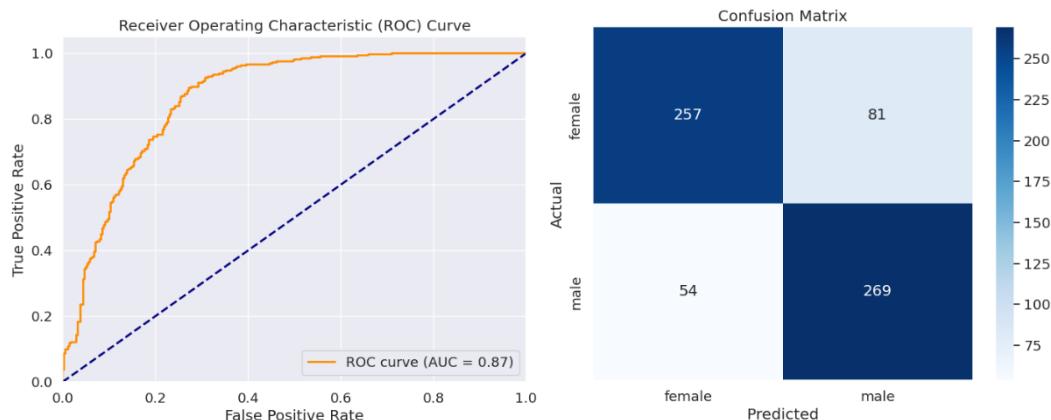
با توجه به ماتریس درهمریختگی شکل ۳۶ متوجه به عملکرد بسیار خوب مدل می‌شویم، همچنین مساحت زیر نمودار AUC برابر با ۰,۹۱ می‌باشد و در ماتریس اشتباه، برای کلاس 'female'، مدل ۲۵۲ بار به درستی و ۸۶ بار به اشتباه پیش‌بینی کرده است، و برای کلاس "male" ۲۸۷ بار به درستی و ۳۶ بار

به اشتباه پیش‌بینی کرده است. این نتایج نشان‌دهنده تعادل خوبی بین دو کلاس هستند، با وجود اینکه تعداد خطاهای کلاس 'female' بیشتر است. همچنین نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۳۷ قابل مشاهده هستند.

Classification Report:					
	precision	recall	f1-score	support	
0	0.88	0.75	0.81	338	
1	0.77	0.89	0.82	323	
accuracy			0.82	661	
macro avg	0.82	0.82	0.81	661	
F1 score:	0.824712643678161	weighted avg	0.82	0.81	661

شکل ۳۷. نتایج ارزیابی کلاسی و کلی داده برای RBF SVM با کاهش بعد PCA و با نرمال‌سازی

#### شکل ۳۸. ماتریس درهم‌بختگی و منحنی ROC برای Logistic Regression-۲-۸-۴



شکل ۳۸. ماتریس درهم‌بختگی و منحنی ROC برای Logistic Regression-۲-۸-۴ با کاهش بعد PCA و با نرمال‌سازی

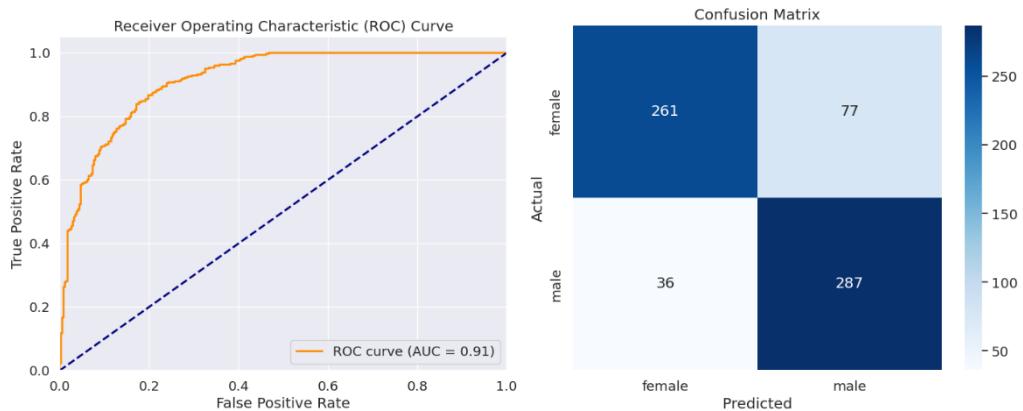
با توجه به ماتریس درهم‌بختگی شکل ۳۸ متوجه به عملکرد خوب مدل می‌شویم، همچنین مساحت زیر نمودار AUC برابر با ۰,۸۷ می‌باشد و PCA در کاهش ابعاد موثر بوده و به نظر می‌رسد که مدل رگرسیون لجستیک به خوبی از داده‌های نرمال‌سازی شده بهره برده است. با این حال، عملکرد این مدل به اندازه برخی از مدل‌های دیگری که دارای AUC بالاتر بودند، نیست. این ممکن است به دلیل ماهیت خطی رگرسیون لجستیک باشد که ممکن است در داده‌هایی که دارای روابط غیرخطی هستند، به خوبی عمل نکند. همچنین نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۳۹ قابل مشاهده هستند.

Classification Report:					
	precision	recall	f1-score	support	
0	0.83	0.76	0.79	338	
1	0.77	0.83	0.80	323	
accuracy			0.80	661	
macro avg	0.80	0.80	0.80	661	
weighted avg	0.80	0.80	0.80	661	

Precision: 0.7685714285714286  
Recall: 0.8328173374613003  
F1 score: 0.799405646359584

شکل ۳۹. نتایج ارزیابی کلاسی و کلی داده برای Logistic Regression با کاهش بعد PCA و با نرمال سازی

#### MLP-۲-۸-۵ با کاهش ابعاد و با نرمال سازی



شکل ۴۰. ماتریس درهم ریختگی و منحنی ROC برای MLP با کاهش بعد PCA و با نرمال سازی

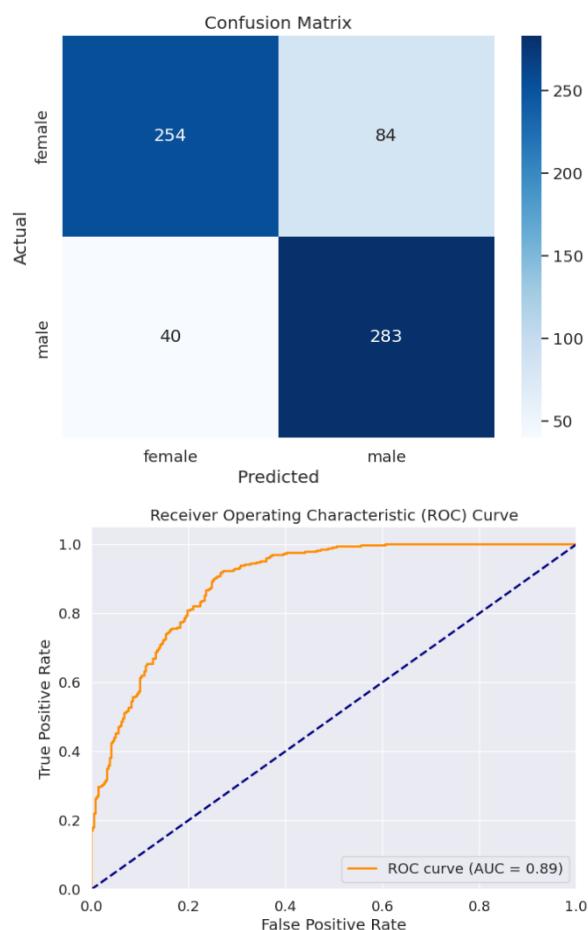
با توجه به ماتریس درهم ریختگی شکل ۴۰، متوجه به عملکرد خوب مدل می شویم، همچنین مساحت زیر نمودار AUC برابر با ۰,۹۱ می باشد ماتریس اشتباه نشان می دهد که مدل برای کلاس 'female' 261 بار به درستی و ۷۷ بار به اشتباه پیش بینی کرده است، و برای کلاس 'male' ، ۲۸۷ بار به درستی و ۳۶ بار به اشتباه پیش بینی کرده است. این نتایج نشان می دهند که مدل در تشخیص هر دو کلاس عملکرد خوبی داشته است، با این حال تعداد خطاهای در پیش بینی کلاس 'female' بیشتر است.

PCA به کاهش ابعاد داده ها کمک کرده و نرمال سازی به استانداردسازی مقیاس ویژگی ها کمک کرده است. مدل MLP معمولاً در پردازش ویژگی های غیر خطی و تعمیم یادگیری به داده های جدید عملکرد خوبی دارد. در این مورد، به نظر می رسد که ترکیب PCA و داده های نرمال سازی شده به همراه شبکه عصبی چند لایه پرسپترون نتایج مطلوبی ارائه داده است. همچنین نتایج ارزیابی داده های تست مربوط به هر کلاس و به صورت کلی در شکل ۴۱ قابل مشاهده هستند.

Classification Report:							
	precision	recall	f1-score	support			
Precision:	0.7884615384615384	0	0.88	0.77	0.82	338	
Recall:	0.8885448916408669	1	0.79	0.89	0.84	323	
F1 score:	0.8355167394468705			accuracy	0.83	661	
			macro avg	0.83	0.83	0.83	661
			weighted avg	0.83	0.83	0.83	661

شکل ۴۱. نتایج ارزیابی کلاسی و کلی داده برای MLP با کاهش بعد PCA و با نرمال سازی

#### ۲-۸-۶ Ensemble Method با کاهش ابعاد و با نرمال سازی



شکل ۴۲. ماتریس درهمریختگی و منحنی ROC برای Ensemble models با کاهش بعد PCA و با نرمال سازی

با توجه به ماتریس درهمریختگی شکل ۴۲ متوجه به عملکرد خوب مدل می‌شویم، همچنین منحنی ROC با AUC (مساحت زیر منحنی) برابر با ۰,۸۹ نشان‌دهنده عملکرد خوبی است. این مقدار نشان می‌دهد که مدل به خوبی توانسته بین دو کلاس تفاوت قائل شود، اگرچه نسبت به برخی مدل‌های دیگری که دارای AUC بالاتر بودند، کمی پایین‌تر است. همچنین نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۴۳ قابل مشاهده هستند.

Classification Report:					
	precision	recall	f1-score	support	
0	0.86	0.75	0.80	338	
1	0.77	0.88	0.82	323	
accuracy			0.81	661	
macro avg	0.82	0.81	0.81	661	
weighted avg	0.82	0.81	0.81	661	

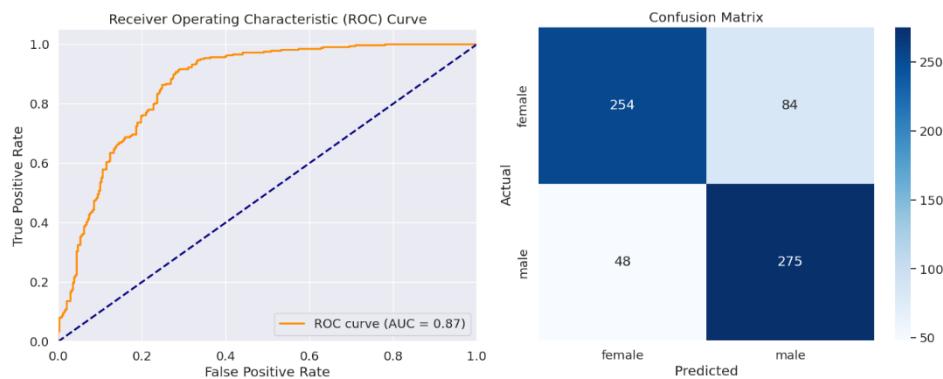
  

Precision: 0.771117166212534
Recall: 0.8761609907120743
F1 score: 0.8202898550724637

شکل ۴۳. نتایج ارزیابی کلاسی و کلی داده برای Ensemble Models با کاهش بعد PCA و بدون نرمال سازی

اجرای شبیه سازی ها بدون نرمال سازی را مجددا برای کاهش ابعادی که توسط PCA دادیم، انجام خواهیم داد که در ادامه بررسی خواهیم نمود.

#### Linear SVM-۲-۸-۷ با کاهش ابعاد و بدون نرمال سازی



شکل ۴۴. ماتریس درهم ریختگی و منحنی ROC برای Linear SVM با کاهش بعد PCA و بدون نرمال سازی

با توجه به ماتریس درهم ریختگی شکل ۴۴ متوجه به عملکرد خوب مدل می شویم، همچنین مساحت زیر نمودار AUC برابر با ۰,۸۷ می باشد و منحنی ROC با AUC (مساحت زیر منحنی) برابر با ۰,۸۷ نشان دهنده عملکرد متوسط تا خوب مدل است. این مقدار کمی کمتر از مدل هایی است که با داده های نرمال سازی شده آموزش دیده اند، نشان دهنده این است که نرمال سازی ممکن است به بهبود توانایی مدل در تشخیص دقیق بین کلاس ها کمک کند. همچنین نتایج ارزیابی داده های تست مربوط به هر کلاس و به صورت کلی در شکل ۴۵ قابل مشاهده هستند.

Classification Report:					
	precision	recall	f1-score	support	
0	0.84	0.75	0.79	338	
1	0.77	0.85	0.81	323	
accuracy			0.80	661	
macro avg	0.80	0.80	0.80	661	
weighted avg	0.80	0.80	0.80	661	

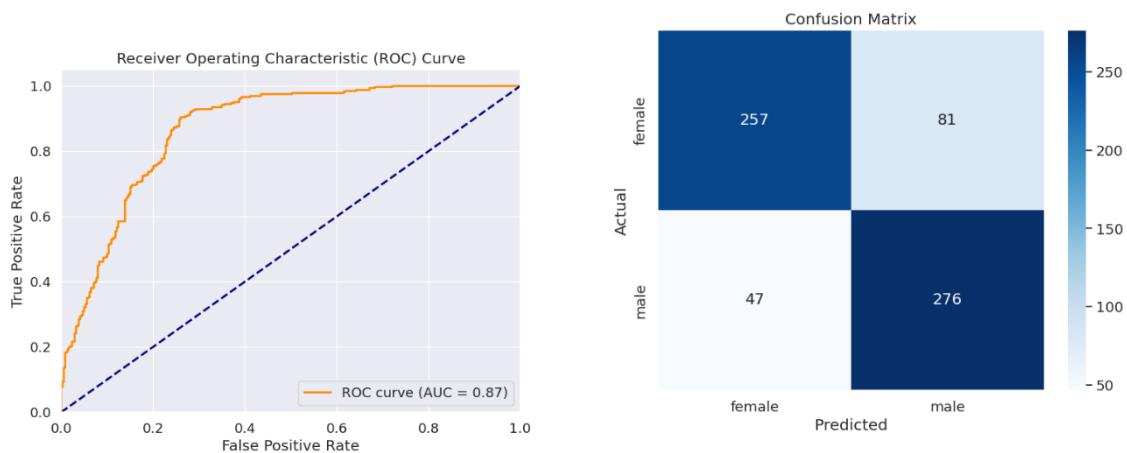
  

Precision: 0.766016713091922
Recall: 0.8513931888544891
F1 score: 0.8064516129032258

شکل ۴۵. نتایج ارزیابی کلاسی و کلی داده برای Linear SVM با کاهش بعد PCA و بدون نرمال سازی

## ۲-۸-۸ باکاهش ابعاد و بدون نرمالسازی Naïve Bayes

این مدل نیز بسیار ساده بوده و با فرض مرتبط نبودن ویژگی‌ها و قانون بیز تلاش بر دسته‌بندی داده‌ها دارد، با این حال این مدل نیز عملکرد نسبتاً مناسبی برای هر دو کلاس داشته است که ماتریس درهمریختگی و منحنی ROC در شکل زیر قابل مشاهده است.



شکل ۴۶. ماتریس درهمریختگی و منحنی ROC برای Naive Bayes با کاهش بعد PCA و بدون نرمالسازی

با توجه به ماتریس درهمریختگی شکل ۴۶، متوجه به عملکرد خوب مدل می‌شویم، همچنین مساحت زیر نمودار AUC برابر با ۰,۸۷ می‌باشد و

تحلیل مؤلفه‌های اصلی به کاهش ابعاد داده‌ها کمک کرده است، اما نتایج نشان می‌دهند که نرمالسازی ممکن است بهبود عملکرد کلی مدل کمک کند، خصوصاً با مدل Naive Bayes که بر پایه احتمالات است و می‌تواند به شدت تحت تأثیر مقیاس ویژگی‌ها قرار گیرد. این نتایج ممکن است بهینه نباشند، و عملکرد مدل با نرمالسازی داده‌ها بهبود پیدا می‌کند.

همچنین نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۴۷ قابل مشاهده هستند.

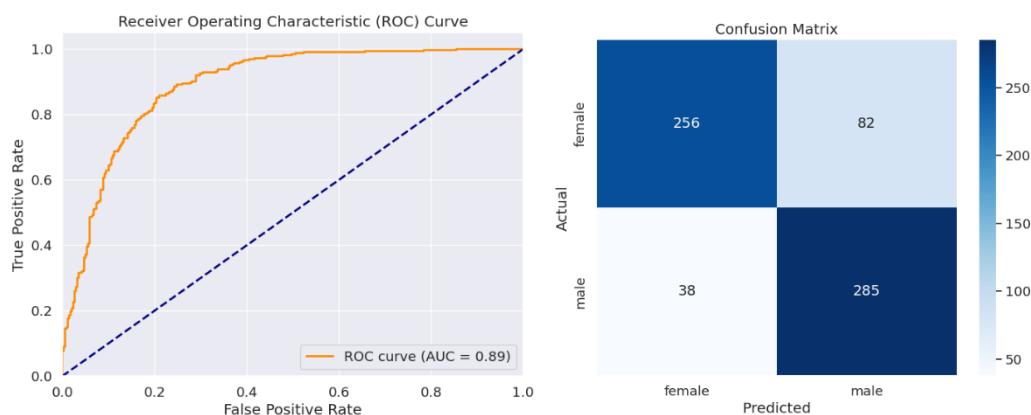
Classification Report:					
	precision	recall	f1-score	support	
0	0.85	0.76	0.80	338	
1	0.77	0.85	0.81	323	
accuracy			0.81	661	
macro avg	0.81	0.81	0.81	661	
weighted avg	0.81	0.81	0.81	661	

Precision: 0.773109243697479  
 Recall: 0.8544891640866873  
 F1 score: 0.811764705882353

شکل ۴۷. نتایج ارزیابی کلاسی و کلی داده برای Naive Bayes با کاهش بعد PCA و بدون نرمال سازی

## RBF SVM-۲-۸-۹

ماتریس درهمریختگی و منحنی ROC در شکل ۴۸ قابل مشاهده است.



شکل ۴۸. ماتریس درهمریختگی و منحنی ROC برای RBF SVM با کاهش بعد PCA و بدون نرمال سازی

با توجه به ماتریس درهمریختگی متوجه به عملکرد بسیار خوب مدل می شویم، همچنین مساحت زیر نمودار AUC برابر با ۰,۸۹ می باشد و ماتریس اشتباه نشان می دهد که مدل برای کلاس 'female' ۲۵۶ بار به درستی و ۸۲ بار به اشتباه پیش بینی کرده است. برای کلاس 'male'، مدل ۲۸۵ بار به درستی و ۳۸ بار به اشتباه پیش بینی کرده است. این نشان دهنده تعداد خطاهای نسبتاً کمتری برای کلاس 'male' است و نشان می دهد که مدل توانسته توازن خوبی بین تشخیص صحیح و خطاهای برای هر دو کلاس حفظ کند. همچنین نتایج ارزیابی داده های تست مربوط به هر کلاس و به صورت کلی در شکل ۴۹ قابل مشاهده هستند.

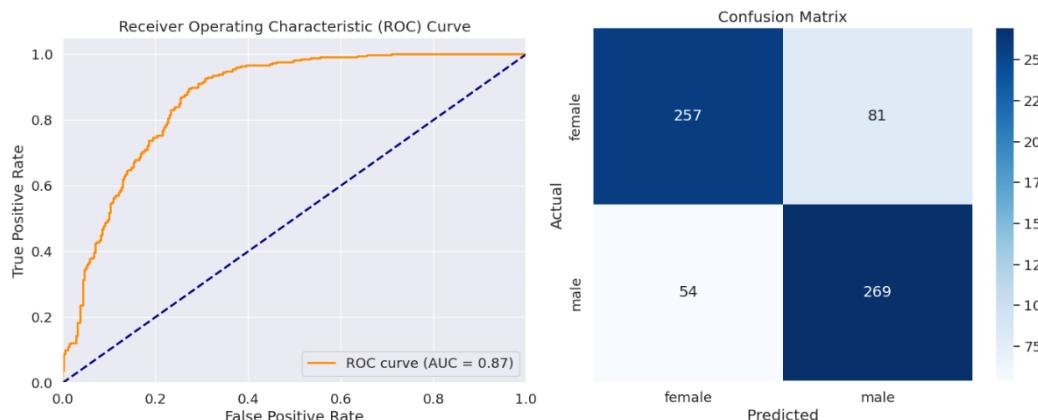
Classification Report:				
	precision	recall	f1-score	support
0	0.87	0.76	0.81	338
1	0.78	0.88	0.83	323
Precision:	0.776566757493188			
Recall:	0.8823529411764706	accuracy	0.82	661
F1 score:	0.8260869565217391	macro avg	0.82	661
		weighted avg	0.82	661

شکل ۴۹. نتایج ارزیابی کلاسی و کلی داده برای RBF SVM با کاهش بعد PCA و بدون نرمال سازی

این مدل از دو مدل قبلی به شدت بهتر عمل کرده است و دلیل آن مشخصاً به خاطر خاصیت غیر خطی می‌باشد.

#### ۲-۸-۱۰ Logistic Regression با کاهش ابعاد و بدون نرمال سازی

این مدل بسیار ساده بوده و فقط قابلیت جداسازی داده‌ها به صورت خطی را دارد می‌باشد، با این حال نتایج نسبتاً مناسبی بر روی داده‌های ما دریافت می‌کند.



شکل ۵۰. ماتریس درهمبریختگی و منحنی ROC برای Logistic Regression با کاهش بعد PCA و بدون نرمال سازی

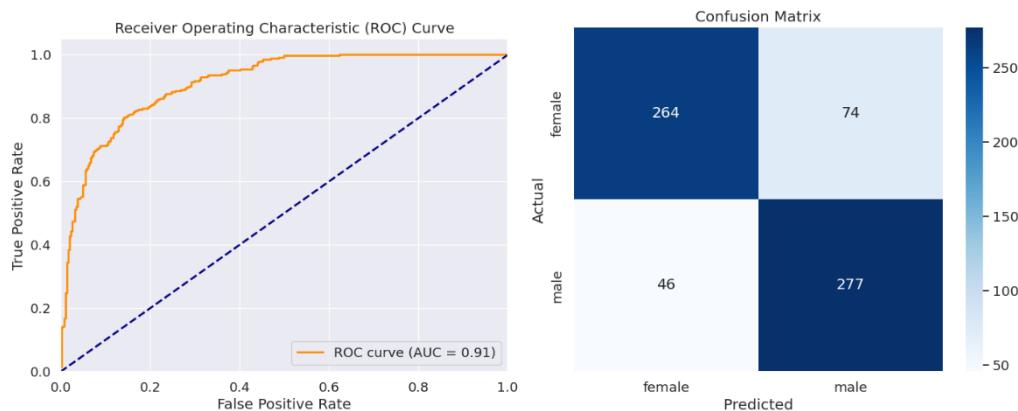
با توجه به ماتریس درهمبریختگی شکل ۵۰، متوجه به عملکرد خوب مدل می‌شویم، همچنان منحنی ROC با AUC (مساحت زیر منحنی) برابر با ۰,۸۷ نشان‌دهنده عملکرد خوبی است، که نسبتاً نزدیک به عملکرد ایده‌آل است اما نشان می‌دهد که هنوز جای بهبود وجود دارد. همچنان نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۵۱ قابل مشاهده هستند.

Classification Report:					
	precision	recall	f1-score	support	
0	0.83	0.76	0.79	338	
	0.77	0.83	0.80	323	
accuracy			0.80	661	
macro avg	0.80	0.80	0.80	661	
weighted avg	0.80	0.80	0.80	661	

Precision: 0.7685714285714286  
 Recall: 0.8328173374613003  
 F1 score: 0.799405646359584

شکل ۵۱. نتایج ارزیابی کلاسی و کلی داده برای Logistic Regression با کاهش بعد PCA و بدون نرمال سازی

## MLP-۲-۸-۱۱ با کاهش ابعاد و بدون نرمال سازی



شکل ۵۲. ماتریس درهمیریختگی و منحنی ROC برای MLP با کاهش بعد PCA و بدون نرمال سازی

با توجه به ماتریس درهمیریختگی شکل ۵۲، متوجه به عملکرد خوب مدل می‌شویم، همچنین منحنی ROC با AUC (مساحت زیر منحنی) برابر با ۰,۹۱ نشان‌دهنده عملکرد خوبی است.

PCA به کاهش ابعاد کمک کرده و ممکن است به مدل اجازه داده باشد تا روی ویژگی‌های مهم‌تر تمرکز کند. با این حال، نتایج نشان می‌دهند که مدل MLP حتی بدون نرمال سازی داده‌ها توانسته است عملکرد خوبی از خود نشان دهد، اگرچه ممکن است نرمال سازی می‌توانست به بهبود دقت کلی مدل کمک کند. شبکه‌های عصبی چند لایه پرسپترون اغلب در پردازش ویژگی‌های غیرخطی و تعمیم یادگیری به داده‌های جدید کارآمد هستند

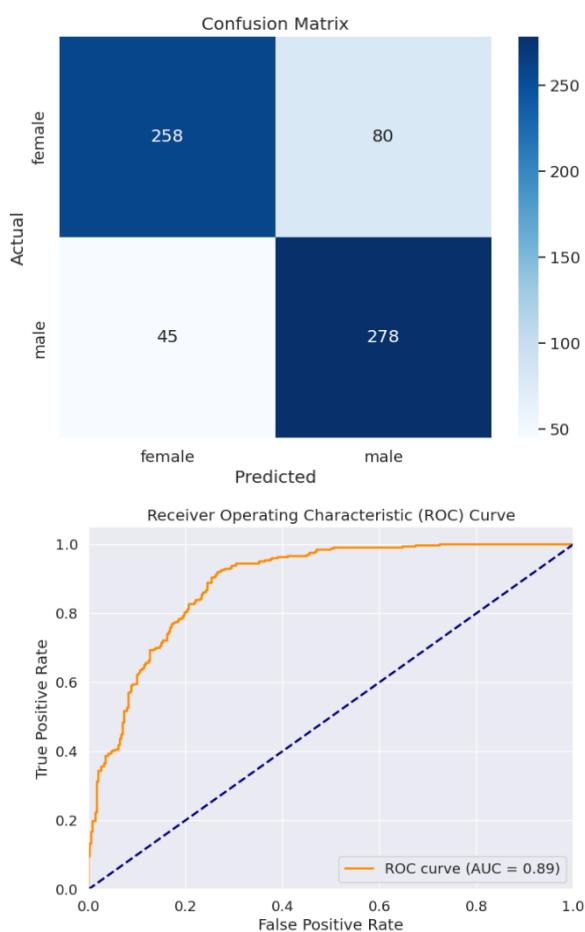
همچنین نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۵۳ قابل مشاهده هستند.

Classification Report:				
	precision	recall	f1-score	support
0	0.85	0.78	0.81	338
1	0.79	0.86	0.82	323
	accuracy		0.82	661
	macro avg	0.82	0.82	661
	weighted avg	0.82	0.82	661

Precision: 0.7891737891737892  
Recall: 0.8575851393188855  
F1 score: 0.8219584569732938

شکل ۵۳. نتایج ارزیابی کلاسی و کلی داده برای MLP با کاهش بعد PCA و بدون نرمال سازی

### Ensemble Method-۲-۸-۱۲



شکل ۵۴. ماتریس درهم ریختگی و منحنی ROC برای Ensemble models با کاهش بعد PCA و بدون نرمال سازی

با توجه به ماتریس درهم ریختگی شکل ۵۴، متوجه به عملکرد خوب مدل می شویم، همچنین مساحت زیر نمودار AUC برابر با ۰,۸۹ می باشد و ماتریس اشتباه نشان می دهد که مدل برای کلاس 'زنانه' ۲۵۸ بار به درستی و ۸۰ بار به اشتباه پیش بینی کرده است. برای کلاس 'مردانه'، مدل ۲۷۸ بار به درستی و ۴۵ بار به اشتباه پیش بینی کرده است. این نتایج نشان دهنده عملکرد متوازن نسبتاً خوبی در پیش بینی هر دو کلاس هستند، با این حال تعداد خطاهای کمی بیشتر برای کلاس 'زنانه' وجود دارد.

استفاده همچنین نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۵۵ قابل مشاهده هستند.

Classification Report:					
	precision	recall	f1-score	support	
0	0.85	0.76	0.80	338	
1	0.78	0.86	0.82	323	
accuracy			0.81	661	
macro avg	0.81	0.81	0.81	661	
weighted avg	0.81	0.81	0.81	661	

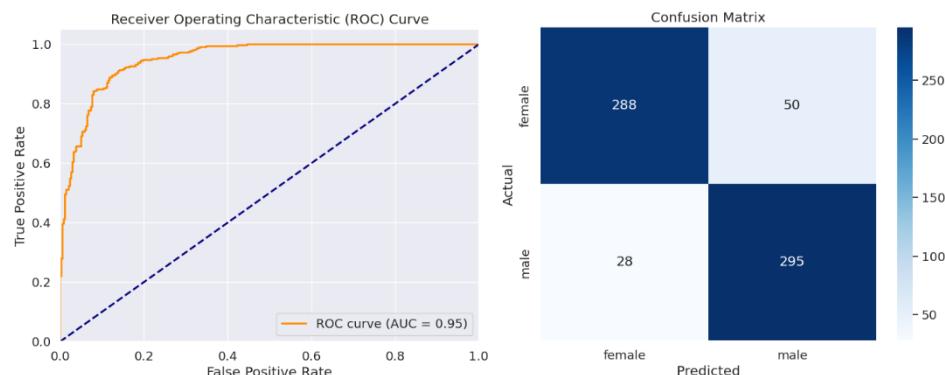
Precision: 0.776536312849162
Recall: 0.8606811145510835
F1 score: 0.8164464023494861

شکل ۵۵. نتایج ارزیابی کلاسی و کلی داده برای Ensemble Models با کاهش بعد PCA و بدون نرمال‌سازی

## ۲-۹-کاهش بعد با استفاده از LDA

فرایند بخش قبل را مجدداً این بار با LDA انجام خواهیم داد. ابتدا با نرمال‌سازی نتایج را ملاحظه خواهیم نمود.

### ۲-۹-۱ با کاهش ابعاد و با نرمال‌سازی Linear SVM



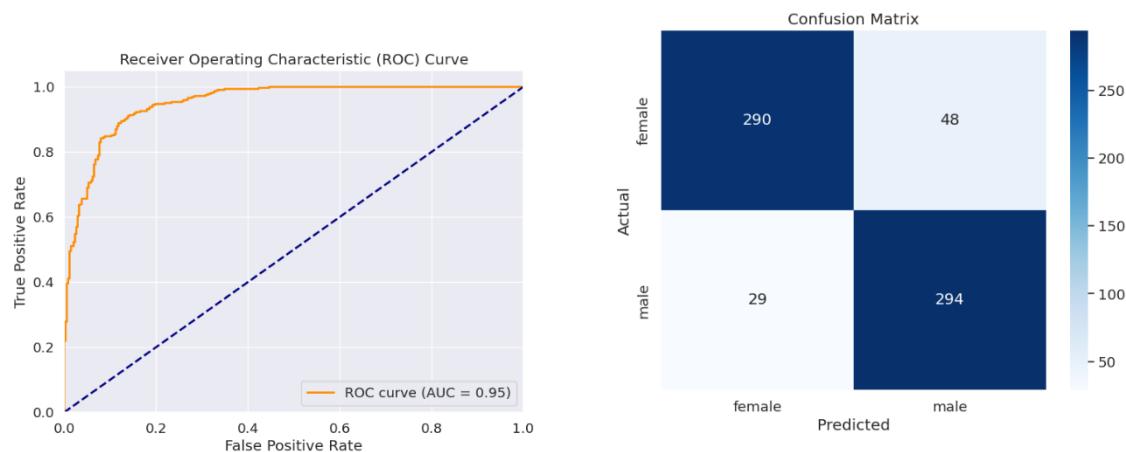
شکل ۵۶. ماتریس درهمیرختگی و منحنی ROC برای Linear SVM با کاهش بعد LDA و با نرمال‌سازی

با توجه به ماتریس درهمیرختگی شکل ۵۶، متوجه به عملکرد خوب مدل می‌شویم، همچنین مساحت زیر نمودار AUC برابر با ۰,۹۵ می‌باشد و اتریس اشتباه نشان می‌دهد که مدل برای کلاس 'female' 288 بار به درستی و ۵۰ بار به اشتباه پیش‌بینی کرده است، و برای کلاس 'male' 295 بار به درستی و ۲۸ بار به اشتباه پیش‌بینی کرده است. این نتایج نشان می‌دهند که مدل عملکرد خوبی در تشخیص هر دو کلاس داشته است، با تعداد نسبتاً کمی از خطاهای همچنین نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۵۷ قابل مشاهده هستند.

Classification Report:					
	precision	recall	f1-score	support	
0	0.91	0.85	0.88	338	Precision: 0.855072463768116
1	0.86	0.91	0.88	323	Recall: 0.913312693498452
accuracy			0.88	661	F1 score: 0.8832335329341316
macro avg	0.88	0.88	0.88	661	
weighted avg	0.88	0.88	0.88	661	

شکل ۵۷. نتایج ارزیابی کلاسی و کلی داده برای Linear SVM با کاهش بعد LDA و با نرمال سازی

## ۲-۹-۲ با کاهش ابعاد و با نرمال سازی Naïve Bayes



شکل ۵۸. ماتریس درهم ریختگی و منحنی ROC برای Naive Bayes با کاهش بعد LDA و با نرمال سازی

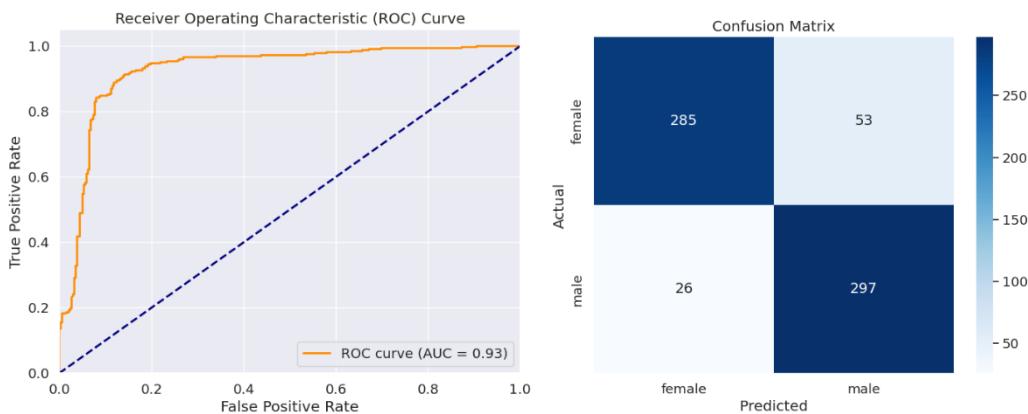
با توجه به ماتریس درهم ریختگی شکل ۵۸ متوجه به عملکرد خوب مدل می شویم، همچنین مساحت زیر نمودار AUC برابر با ۰,۹۵ می باشد این مدل از مدل SVM بدتر عمل کرده است. LDA به عنوان یک روش کاهش بعد عمل کرده و به مدل کمک کرده است تا روی جنبه های مهم تر داده ها تمرکز کند که برای تفکیک کلاس ها موثر هستند. همچنین نتایج ارزیابی داده های تست مربوط به هر کلاس و به صورت کلی در شکل ۵۹ قابل مشاهده هستند.

Classification Report:					
	precision	recall	f1-score	support	
Precision: 0.8596491228070176	0	0.91	0.86	0.88	338
Recall: 0.9102167182662538	1	0.86	0.91	0.88	323
F1 score: 0.8842105263157896	accuracy			0.88	661
	macro avg	0.88	0.88	0.88	661
	weighted avg	0.88	0.88	0.88	661

شکل ۵۹. نتایج ارزیابی کلاسی و کلی داده برای Naive Bayes با کاهش بعد LDA و با نرمال سازی

### RBF SVM-۲-۹-۳ با کاهش ابعاد و با نرمال سازی

ماتریس در هم ریختگی و منحنی ROC در شکل ۶۰ قابل مشاهده است.



شکل ۶۰. ماتریس در هم ریختگی و منحنی ROC برای RBF SVM با کاهش بعد LDA و با نرمال سازی

با توجه به ماتریس در هم ریختگی شکل ۶۰ متوجه به عملکرد بسیار خوب مدل می شویم، همچنین مساحت زیر نمودار AUC برابر با  $0.93$  می باشد و مدل در هر دو کلاس مرد و زن تقریباً به یک شکل عمل کرده است. همچنین نتایج ارزیابی داده های تست مربوط به هر کلاس و به صورت کلی در شکل ۶۱ قابل مشاهده هستند.

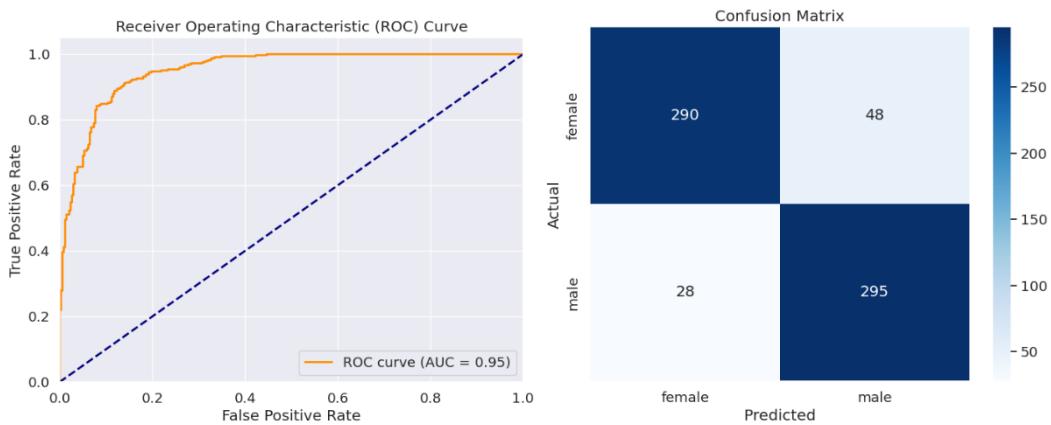
Classification Report:					
	precision	recall	f1-score	support	
0	0.92	0.84	0.88	338	
1	0.85	0.92	0.88	323	
Precision: 0.8485714285714285					accuracy
Recall: 0.9195046439628483					macro avg
F1 score: 0.8826151560178306					weighted avg

شکل ۶۱. نتایج ارزیابی کلاسی و کلی داده برای RBF SVM با کاهش بعد LDA و با نرمال سازی

این مدل از دو مدل قبلی به شدت بهتر عمل کرده است و دلیل آن مشخصاً به خاطر خاصیت غیر خطی می باشد.

### Logistic Regression -۲-۹-۴ با کاهش ابعاد و با نرمال سازی

این مدل بسیار ساده بوده و فقط قابلیت جداسازی داده ها به صورت خطی را دارا می باشد، با این حال نتایج نسبتاً مناسبی بر روی داده های ما دریافت می کند.



شکل ۶۲. ماتریس درهمیریختگی و منحنی ROC برای Logistic Regression با کاهش بعد LDA و با نرمال سازی

با توجه به ماتریس درهمیریختگی شکل ۶۲، متوجه به عملکرد خوب مدل می‌شویم، همچنین مساحت زیر نمودار AUC برابر با  $0.95$  می‌باشد و ماتریس اشتباه نشان می‌دهد که مدل در تشخیص کلاس 'female' ۲۸۵ بار موفق بوده و ۵۳ بار نادرست پیش‌بینی کرده است. برای کلاس 'male'، مدل ۲۹۷ بار به درستی و ۲۶ بار به اشتباه تشخیص داده است. این نتایج نشان‌دهنده توانایی مدل در تشخیص دقیق هر دو کلاس با تعداد خطاهای نسبتاً کم. همچنین نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۶۳ قابل مشاهده هستند.

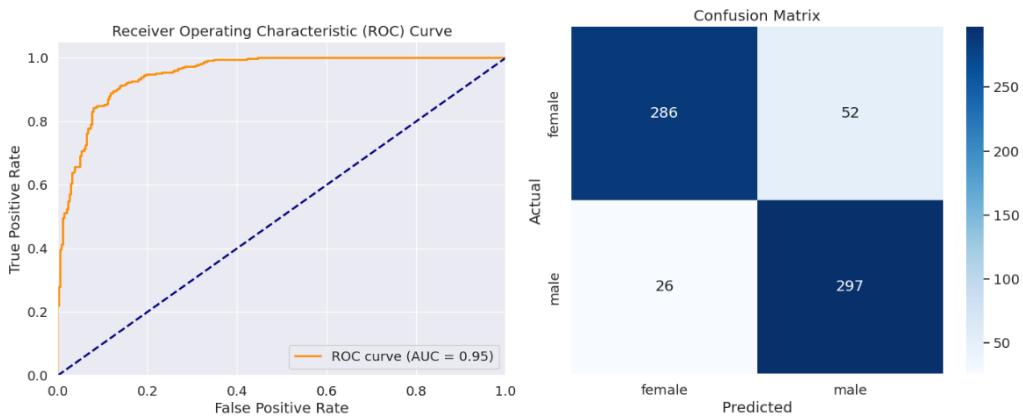
Classification Report:				
	precision	recall	f1-score	support
0	0.91	0.86	0.88	338
1	0.86	0.91	0.89	323
accuracy			0.89	661
macro avg	0.89	0.89	0.89	661
weighted avg	0.89	0.89	0.88	661

Precision: 0.8600583090379009  
Recall: 0.913312693498452  
F1 score: 0.8858858858858859

شکل ۶۳. نتایج ارزیابی کلاسی و کلی داده برای Logistic Regression با کاهش بعد LDA و با نرمال سازی

### MLP-۲-۹-۵ با کاهش ابعاد و با نرمال سازی

شبکه‌ی عصبی چند لایه قابلیت، دسته‌بندی داده‌ها به شدت پیچیده را دارد، لذا ما توقع داریم این مدل بهترین عملکرد را نسبت به سایر مدل‌ها به ما بدهد.



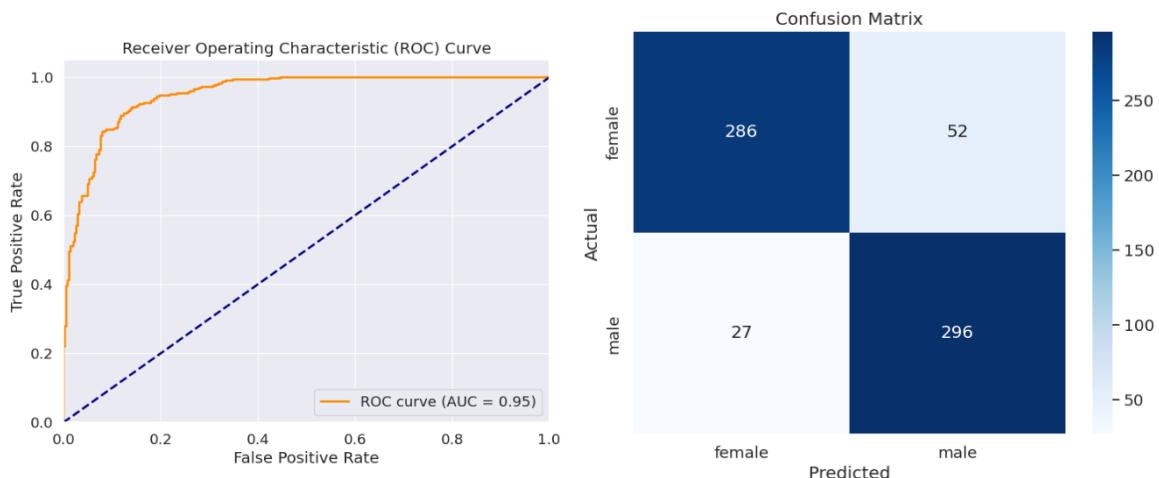
شکل ۶۴. ماتریس درهمریختگی و منحنی ROC برای MLP با کاهش بعد LDA و با نرمال سازی

با توجه به ماتریس درهمریختگی شکل ۶۴، متوجه به عملکرد خوب مدل می‌شویم، همچنین مساحت زیر نمودار AUC برابر با  $0.95 \pm 0.05$  می‌باشد. همچنین نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل ۶۵ قابل مشاهده هستند.

Classification Report:					
	precision	recall	f1-score	support	
Precision: 0.8510028653295129	0	0.92	0.85	0.88	338
Recall: 0.9195046439628483	1	0.85	0.92	0.88	323
F1 score: 0.8839285714285715			accuracy	0.88	661
			macro avg	0.88	661
			weighted avg	0.88	661

شکل ۶۵. نتایج ارزیابی کلاسی و کلی داده برای MLP با کاهش بعد LDA و با نرمال سازی

#### با کاهش ابعاد و با نرمال سازی Ensemble Method - ۲-۹-۶



شکل ۶۶. ماتریس درهمریختگی و منحنی ROC برای Ensemble models با کاهش بعد LDA و با نرمال سازی

مساحت زیر نمودار AUC در شکل ۶۶، برابر با ۰,۹۵ می باشد و LDA به کاهش بُعد دادهها کمک کرده و به مدل اجازه داده است تا روی جنبه های مهم تری که برای تفکیک کلاس ها حیاتی هستند تمرکز کند. نرمال سازی داده ها نیز اطمینان می دهد که ویژگی ها در مقیاس های یکسانی قرار می گیرند، که به افزایش دقیق مدل کمک می کند. همچنین نتایج ارزیابی داده های تست مربوط به هر کلاس و به صورت کلی در شکل ۶۷ قابل مشاهده هستند.

Classification Report:					
	precision	recall	f1-score	support	
0	0.91	0.85	0.88	338	
1	0.85	0.92	0.88	323	
accuracy			0.88	661	
macro avg	0.88	0.88	0.88	661	
weighted avg	0.88	0.88	0.88	661	

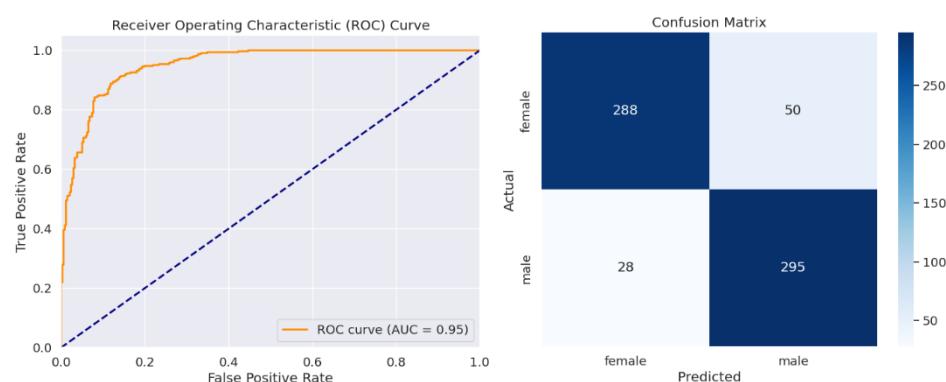
  

Precision: 0.8505747126436781
Recall: 0.9164086687306502
F1 score: 0.8822652757078987

شکل ۶۷. نتایج ارزیابی کلاسی و کلی داده برای Ensemble Models با کاهش بعد LDA و با نرمال سازی

حال مجدداً روند قبلی را بدون استفاده از نرمال سازی تکرار خواهیم کرد.

#### ۲-۹-۷ با کاهش ابعاد و بدون نرمال سازی Linear SVM



شکل ۶۸. ماتریس در هم ریختگی و منحنی ROC برای Linear SVM با کاهش بعد LDA و بدون نرمال سازی

مساحت زیر نمودار AUC برابر با ۰,۹۵ است. همچنین نتایج ارزیابی داده های تست مربوط به هر کلاس و به صورت کلی در شکل زیر قابل مشاهده هستند.

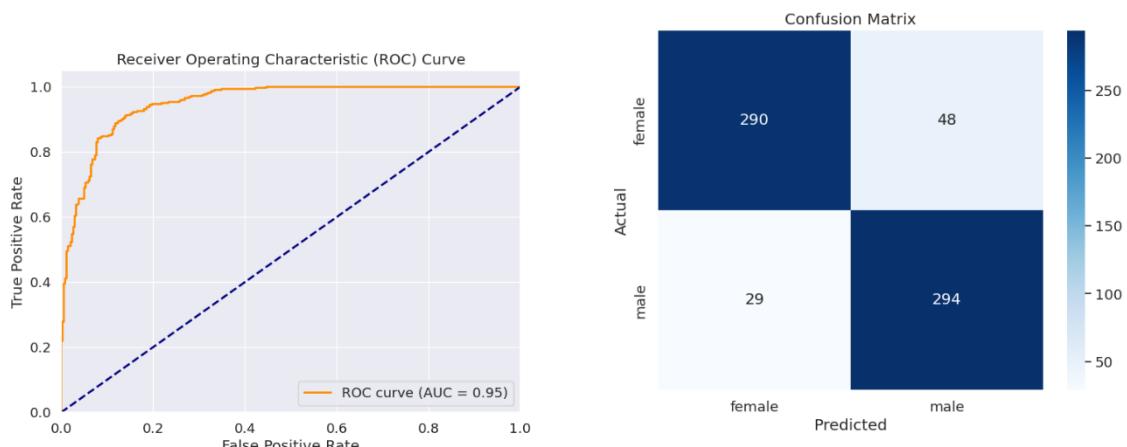
مقادیر دقیق، بازیابی و مقدار F1 برای هر کلاس در هر دو گزارش نزدیک به هم هستند، که نشان دهنده تعادل و یکنواختی در عملکرد مدل ها است. عموماً، دقت بالای ۰,۸۵، بازیابی بالای ۰,۹۱ و مقدار F1 بالای

۶۸ در هر دو گزارش نشان‌دهنده عملکرد نسبتاً قوی مدل‌ها در دسته‌بندی است. این نوع اطلاعات به ارزیابی دقیق‌تر مدل‌ها کمک می‌کند و درک بهتری از قدرت و ضعف‌های آن‌ها در پیش‌بینی کلاس‌های مختلف فراهم می‌آورد.

Classification Report:					Precision: 0.855072463768116
	precision	recall	f1-score	support	Recall: 0.913312693498452
0	0.91	0.85	0.88	338	F1 score: 0.8832335329341316
1	0.86	0.91	0.88	323	
accuracy			0.88	661	
macro avg	0.88	0.88	0.88	661	
weighted avg	0.88	0.88	0.88	661	

شکل ۶۹. نتایج ارزیابی کلاسی و کلی داده برای Linear SVM با کاهش بعد LDA و بدون نرمال‌سازی

## ۲-۹-۸ Naïve Bayes با کاهش ابعاد و بدون نرمال‌سازی



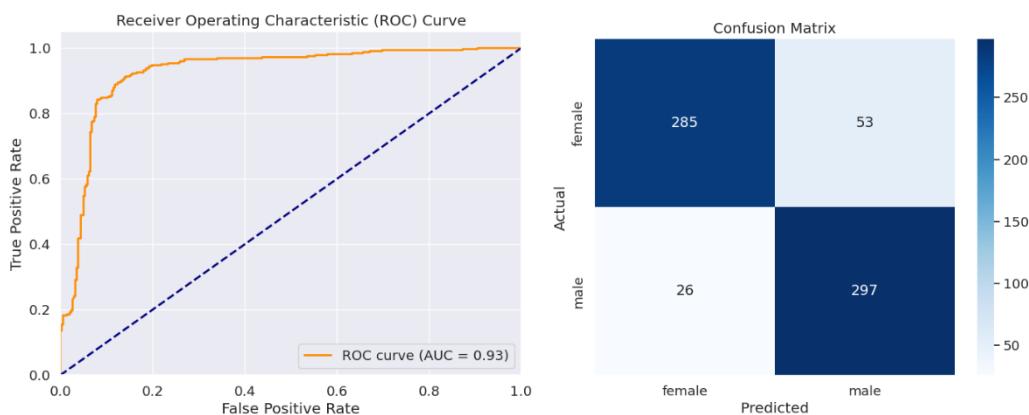
شکل ۷۰. ماتریس درهم‌بختی و منحنی ROC برای Naive Bayes با کاهش بعد LDA و بدون نرمال‌سازی

مساحت زیر نمودار AUC برابر با ۰,۹۵ می‌باشد و مدل در هر دو کلاس مرد و زن تقریباً به یک شکل عمل کرده است. نتایج ارزیابی داده‌های تست مربوط به هر کلاس و به صورت کلی در شکل زیر قابل مشاهده هستند.

Classification Report:					
	precision	recall	f1-score	support	
Precision:	0.8596491228070176				
Recall:		0.9102167182662538			
F1 score:	0.8842105263157896				
		accuracy		0.88	661
		macro avg	0.88	0.88	661
		weighted avg	0.88	0.88	661

شکل ۷۱. نتایج ارزیابی کلاسی و کلی داده برای Naive Bayes با کاهش بعد LDA و بدون نرمال سازی

## RBF SVM - ۲-۹-۹



شکل ۷۲. ماتریس درهم ریختگی و منحنی ROC RBF SVM با کاهش بعد LDA و بدون نرمال سازی

مساحت زیر نمودار AUC برابر با ۰,۹۳ می باشد. همچنین نتایج ارزیابی داده های تست مربوط به هر کلاس و به صورت کلی در شکل زیر قابل مشاهده هستند. دقت برای کلاس ۰ برابر با ۰,۹۲ و برای کلاس ۱ برابر با ۰,۸۵ است. این معیار نشان دهنده نسبت تعداد پیش بینی های صحیح مثبت به کل پیش بینی های مثبت است.

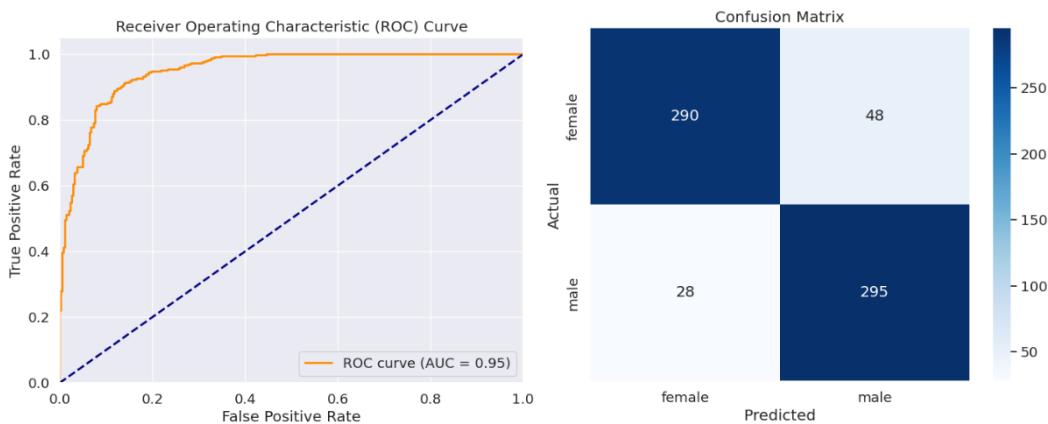
Classification Report:					
	precision	recall	f1-score	support	
Precision:	0.8485714285714285				
Recall:		0.9195046439628483			
F1 score:	0.8826151560178306				
		accuracy		0.88	661
		macro avg	0.88	0.88	661
		weighted avg	0.88	0.88	661

شکل ۷۳. نتایج ارزیابی کلاسی و کلی داده برای RBF SVM با کاهش بعد LDA و بدون نرمال سازی

این مدل از دو مدل قبلی به شدت بهتر عمل کرده است و دلیل آن مشخصا به خاطر خاصیت غیر خطی می باشد.

## با کاهش ابعاد و بدون نرمال سازی Logistic Regression - ۲-۹-۱۰

این مدل بسیار ساده بوده و فقط قابلیت جداسازی داده ها به صورت خطی را دارد می باشد، با این حال نتایج نسبتاً مناسبی بر روی داده های ما دریافت می کند.



شکل ۷۴. ماتریس درهمیریختگی و منحنی ROC برای Logistic Regression با کاهش بعد LDA و بدون نرمال سازی

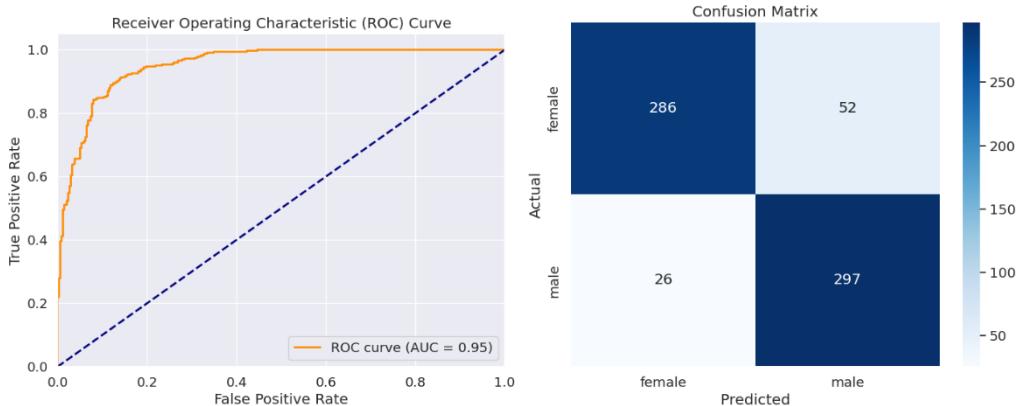
مساحت زیر نمودار AUC برابر با ۰,۹۵ می باشد و مدل در هر دو کلاس مرد و زن تقریباً به یک شکل عمل کرده است. همچنین نتایج ارزیابی داده های تست مربوط به هر کلاس و به صورت کلی در شکل زیر قابل مشاهده هستند.

در مجموع، دقت کلی مدل (accuracy) برابر با ۰,۸۹ است که نشان دهنده عملکرد خوب مدل در دسته بندی است. میانگین ماکرو (macro avg) و میانگین وزن دار (weighted avg) برای همه معیارها نیز برابر با ۰,۸۹ است، که بیانگر تعادل خوب مدل در تشخیص هر دو کلاس است.

Classification Report:					
	precision	recall	f1-score	support	
Precision:	0.8600583090379009	0.91	0.86	0.88	338
Recall:	0.913312693498452	0.86	0.91	0.89	323
F1 score:	0.8858858858858859				
	accuracy			0.89	661
	macro avg	0.89	0.89	0.89	661
	weighted avg	0.89	0.89	0.88	661

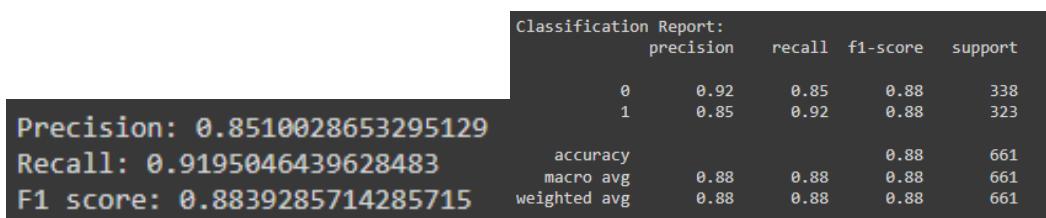
شکل ۷۵. نتایج ارزیابی کلاسی و کلی داده برای Logistic Regression با کاهش بعد LDA و بدون نرمال سازی

## MLP - ۲-۹-۱۱ با کاهش ابعاد و بدون نرمال سازی



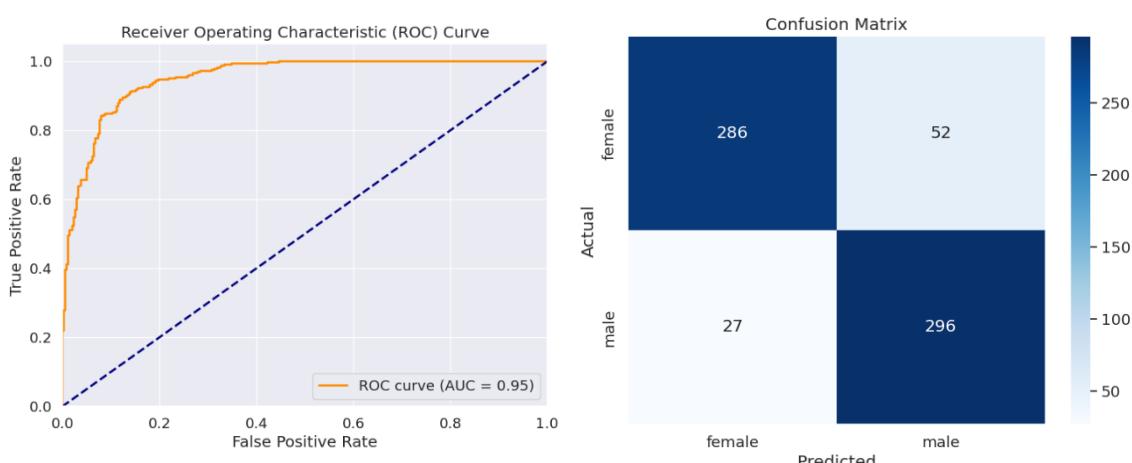
شکل ۷۶. ماتریس درهMRIختگی و منحنی ROC برای MLP با کاهش بعد LDA و بدون نرمال سازی

مساحت زیر نمودار AUC برابر با ۰,۹۵ می باشد. همچنین نتایج ارزیابی داده های تست مربوط به هر کلاس و به صورت کلی در شکل زیر قابل مشاهده هستند. F1 Score میانگین هارمونیک دقت و بازیابی است که یک معیار متوازن بین دقت و بازیابی ارائه می دهد. نمره F1 برای کلاس ۰ برابر با ۰,۸۸ و برای کلاس ۱ برابر با ۰,۸۸ است.



شکل ۷۷. نتایج ارزیابی کلاسی و کلی داده برای MLP با کاهش بعد LDA و بدون نرمال سازی

## Ensemble Method - ۲-۹-۱۲ با کاهش ابعاد و بدون نرمال سازی



شکل ۷۸. ماتریس درهMRIختگی و منحنی ROC برای Ensemble models با کاهش بعد LDA و بدون نرمال سازی

مساحت زیر نمودار AUC برابر با  $0,95$  می باشد. همچنین نتایج ارزیابی داده های تست مربوط به هر کلاس و به صورت کلی در شکل زیر قابل مشاهده هستند.

بازیابی برای کلاس  $0$  برابر با  $0,85$  و برای کلاس  $1$  برابر با  $0,92$  است. این معیار نشان دهنده توانایی مدل در شناسایی کلیه نمونه های مثبت واقعی است.

Classification Report:					
	precision	recall	f1-score	support	
0	0.91	0.85	0.88	338	
1	0.85	0.92	0.88	323	
accuracy			0.88	661	
macro avg	0.88	0.88	0.88	661	
weighted avg	0.88	0.88	0.88	661	

Precision: 0.8505747126436781  
 Recall: 0.9164086687306502  
 F1 score: 0.8822652757078987

شكل ۷۹. نتایج ارزیابی کلاسی و کلی داده برای Ensemble Models با کاهش بعد LDA و بدون نرمال سازی

## ۲-۹-نتیجه گیری و جمع بندی

نتایج را به صورت خلاصه در جدول ۱ می توانید ببینید:

جدول ۱. خلاصه نتایج و مقایسه مدل ها با کاهش ابعاد و بدون کاهش ابعاد و همراه با نرمال سازی و عدم نرمال سازی

	Normalized			Not Normalized		
	Original dimensions	PCA	LDA	Original dimensions	PCA	LDA
<b>Linear SVM</b>	95%	84%	95%	94%	87%	95%
<b>Naïve Bayes</b>	84%	88%	95%	87%	87%	95%
<b>RBF SVM</b>	98%	91%	93%	95%	89%	95%
<b>Logistic Regression</b>	94%	87%	95%	94%	87%	95%
<b>MLP</b>	98%	91%	95%	97%	91%	95%
<b>Ensemble Model</b>	97%	89%	95%	97%	89%	95%

بهترین نتیجه برای SVM با کرنل RBF و MLP با دیتای نرمال سازی شده بود. همچنین مشخص است که PCA نتایج بدتری نسبت به LDA گرفته است و کاهش بعد PCA که به صورت unsupervised انجام می شود بهترین ویژگی ها را جداسازی نکرده است.

با توجه به این جدول، می توان توجه به نکات زیر داشت:

۱. در بیشتر موارد، مدل‌های آموزش دیده بر روی ابعاد اصلی (Original dimensions) نتایج بهتری نسبت به PCA و LDA دارند.
۲. استفاده از نرمالسازی (Normalized) در برخی موارد باعث بهبود دقت مدل‌ها شده است. برای مثال، در MLP و Naïve Bayes با نرمالسازی دقت بیشتری به دست آمده است.
۳. بین الگوریتم‌های مختلف یادگیری ماشینی، هیچ الگوریتم خاصی به طور قطعی بهتر از دیگران عمل نمی‌کند. در هر حالت، نتایج تقریباً مشابه بوده و تفاوت‌ها بسیار کوچک است.
۴. Ensemble Model، که یک ترکیب از مدل‌ها است، در برخی حالات به دقت بیشتری دست یافته است. این نشان می‌دهد که ترکیب چند مدل ممکن است بهبود قابل توجهی در تشخیص جنسیت داشته باشد

## ۳-خوشه‌بندی

### ۱-مقدمه

در این بخش قصد داریم به خوشه بندی دیتاست معرفی شده بپردازیم. دو روش مختلف خوشه بندی که در این قسمت پیاده‌سازی شده‌اند، روش K-Means و روش Spectral Clustering هستند.

خوشه بندی یا به عبارتی Clustering وظیفه گروه بندی مجموعه ای از اشیاء است به گونه ای که اشیاء در یک گروه (به نام خوشه) شباهت بیشتری به یکدیگر داشته باشند تا در گروه های دیگر.

### ۲-K-means Algorithm

به طور کلی در K-means هدف این است که مرکزهایی را انتخاب کند که اینرسی، یا مجموع مربعات درون خوشه را به حداقل برساند:

$$\sum_{i=0}^n \min_{\mu_j \in C} (\|x_i - \mu_j\|^2)$$

اینرسی را می توان به عنوان معیاری برای میزان انسجام درونی خوشه ها درنظر گرفت. از معايib مختلفی تاثير می پذيرد:

۱. اينرسى اين فرض را ايجاد مى کند که خوشه ها محدب و همسانگرد هستند، که هميسه اينطور نیست. به خوشه هاي دراز يا منيفولدهاي با اشكال نامنظم واكنش ضعيفي نشان مى دهد.

۲. اينرسى يك متريک نرمال شده نیست: ما فقط مى دانيم که مقادير کمتر بهتر است و صفر بهينه خواهد بود. اما در فضاهای با ابعاد بسیار بالا، فواصل اقلیدسی تمایل دارند که متورم شوند (این نمونه ای از به اصطلاح "نفرین ابعاد" است). اجرای يك الگوريتم کاهش ابعاد مانند تجزيه و تحليل مؤلفه اصلی (PCA) قبل از خوشه بندی K-means می تواند این مشکل را کاهش دهد و محاسبات را سرعت بخشد.

در پياده سازی K-means سه مرحله وجود دارد:

اولين مرحله، مرکزهای اولیه را انتخاب می کند، با اساسی ترين روش، انتخاب تصادفي پس از مقداردهی اولیه، K-means شامل حلقه زدن بين دو مرحله دیگر است:

۱. مرحله اول هر نمونه را به تزدیکترین مرکز خود اختصاص می دهد.
۲. مرحله دوم با در نظر گرفتن مقدار میانگین تمام نمونه های اختصاص داده شده به هر مرکز قبلی، مرکزهای جدید ایجاد می کند. تفاوت بین مرکز قدیمی و جدید محاسبه می شود و الگوریتم این دو مرحله آخر را تا زمانی که این مقدار کمتر از یک آستانه باشد تکرار می کند. به عبارت دیگر، آنقدر تکرار می شود که مرکزها حرکت قابل توجهی نداشته باشند.

در بخش های بعدی به مراحلی که برای پیاده سازی الگوریتم معرفی شده نیاز است خواهیم پرداخت.

### ۳-۳-پردازش داده ها

بخش ابتدایی کد شامل مراحل آماده سازی دیتابست می باشد، در اینجا همانند قسمت طبقه بندی، ابتدا ماتریس ویژگی ها (features) و سپس لیبل ها را تشکیل می دهیم و به نرمالیزه کردن دادگان می پردازیم. از قطعه کدهایی که در ادامه آوردهیم برای این بخش استفاده نمودیم.

```
First: Loading Data

[ ] #importing necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, silhouette_samples
import matplotlib.cm as cm
from sklearn.decomposition import PCA
import time
import sklearn.cluster as cluster
from sklearn.cluster import AgglomerativeClustering
from matplotlib import style
from sklearn.metrics import silhouette_score
from sklearn.datasets import make_blobs

[ ] from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

[ ] # Specify the path to your NPZ file
file_path = '/content/drive/MyDrive/ML_Project_ASR/mfccs.npy'

data = np.load(file_path, allow_pickle=True)
```

```

▶ # Specify the path to your NPZ file
file_path = '/content/drive/MyDrive/ML_Project_ASR/mfccs.npz'

data = np.load(file_path, allow_pickle=True)

[ ] data.shape
(6042,)

[ ] mean_values_list=[]
for element in data:
    # Calculate the mean along the second axis (axis=1)
    mean_element = np.mean(element, axis=1)

    # Append the result to the new list
    mean_values_list.append(mean_element)

[ ] features=np.array(mean_values_list)
df=pd.read_csv("/content/drive/MyDrive/ML_Project_ASR/cleaned_dataset.csv")
y=df["gender"].values

[ ] #changing labels from categorical to numerical
y[y=="male"]=0
y[y=="female"]=1

[ ] y=np.array(y.tolist())

[ ] print(features.shape)
print(y.shape)

(6042, 13)
(6042,)

[ ] # Normalizing data
min_max_scaler = preprocessing.MinMaxScaler()
X = min_max_scaler.fit_transform(features)

```

در گام بعد با استفاده از روش K-means به خوش بندی دادگان آماده شده اهتمام می‌ورزیم.

### ۴-۳-پیاده‌سازی الگوریتم K-Means و اجرای شبیه‌سازی

برای به دست آوردن تعداد مناسب خوش‌ها آنالیز silhouette را انجام می‌دهیم. در اینجا لازم است دقیق شود که ابتدا صرفا silhouette score را برای تعداد مختلف خوش‌ها به دست آورده و روی یک نمودار رسم کرده‌ایم، و سپس جهت اطمینان بیشتر آنالیز کامل silhouette را انجام داده‌ایم.

کد نوشته شده برای محاسبه مقادیر مختلف silhouette score به صورت زیر می‌باشد:

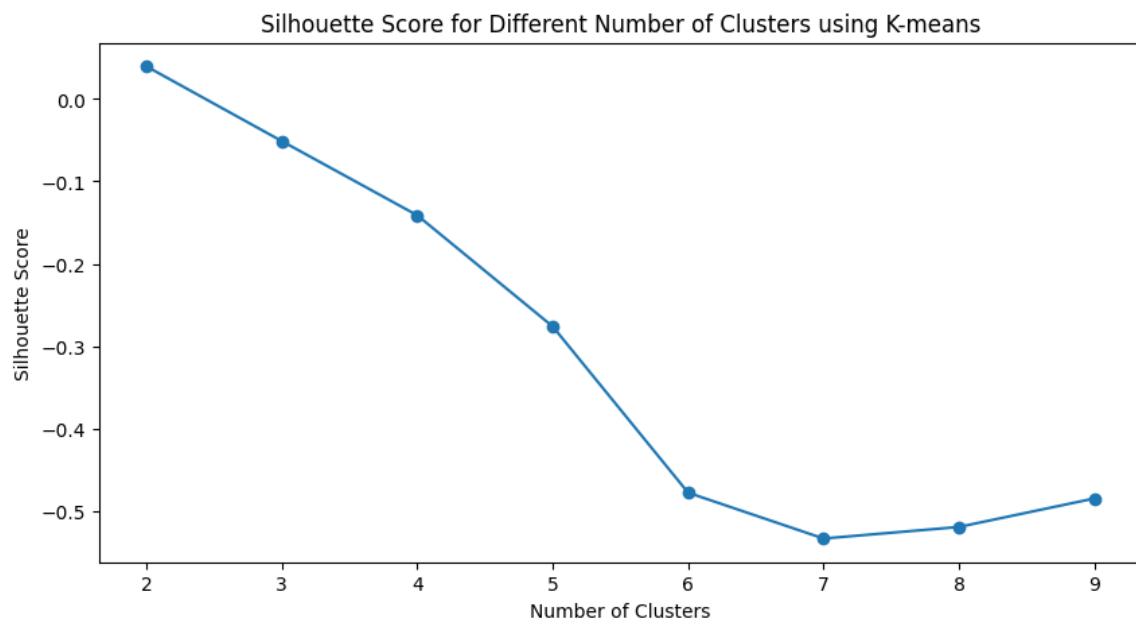
```
Second: K-Means Clustering

168 #plotting silhouette score for different number of clusters
possible_clusters = range(2, 10)
silhouette_scores = []
custom_metrics = []
for n_clusters in possible_clusters:
    kmeans = KMeans(n_clusters=n_clusters, random_state=42)
    cluster_labels = kmeans.fit_predict(X)

    silhouette_avg = silhouette_score(y.reshape(-1, 1), cluster_labels)
    silhouette_scores.append(silhouette_avg)

plt.figure(figsize=(10, 5))
plt.plot(possible_clusters, silhouette_scores, marker='o')
plt.title('Silhouette Score for Different Number of Clusters using K-means')
plt.xlabel('Number of Clusters')
plt.ylabel('Silhouette Score')
plt.show()
```

نمودار حاصل شده مطابق زیر و در شکل ۸۰ آورده شده است.



شکل ۸۰. محاسبه مقادیر silhouette score برای تعداد کلاسترها در الگوریتم K-means

برای مشخص شدن تعداد مناسب خوشه‌ها توجه به این نکته ضروریست که مقدار silhouette score باید عددی مثبت باشد و در موارد منفی به آن معناست که misclassification رخ داده است. لذا در اینجا بهترین تعداد خوشه‌ها برای دادگان معرفی شده برابر با دو خوشه خواهد بود. (زیرا آن عددی بزرگتر از صفر است)

حال برای اطمینان بیشتر و همچنین برای تعمق در بحث خوشه‌بندی، آنالیز silhouette را به طور کامل برای تعداد خوشه‌های ۲ و ۴ و ۶ و ۸ پیاده سازی کرده و نتایج را تفسیر می‌نمائیم.

کد نوشته شده برای این قسمت به صورت زیر خواهد بود:

```
range_n_clusters = [2,4,6,8]

for n_clusters in range_n_clusters:
    # Create a subplot with 1 row and 2 columns
    fig, (ax1, ax2) = plt.subplots(1, 2)
    fig.set_size_inches(18, 7)

    # The 1st subplot is the silhouette plot
    # The silhouette coefficient can range from -1, 1
    ax1.set_xlim([-0.1, 1])
    # The (n_clusters+1)*10 is for inserting blank space between silhouette
    # plots of individual clusters, to demarcate them clearly.
    ax1.set_ylim([0, len(X) + (n_clusters + 1) * 10])

    # Initialize the clusterer with n_clusters value and a random generator
    # seed of 10 for reproducibility.
    clusterer = KMeans(n_clusters=n_clusters, random_state=10)
    cluster_labels = clusterer.fit_predict(X)

    # The silhouette_score gives the average value for all the samples.
    # This gives a perspective into the density and separation of the formed
    # clusters
    silhouette_avg = silhouette_score(X, cluster_labels)
    print("For n_clusters =", n_clusters,
          "The average silhouette_score is :", silhouette_avg)

    # Compute the silhouette scores for each sample
    sample_silhouette_values = silhouette_samples(X, cluster_labels)

    y_lower = 10
```

```

    ith_cluster_silhouette_values.sort()

    size_cluster_i = ith_cluster_silhouette_values.shape[0]
    y_lower = y_lower + size_cluster_i

    color = cm.nipy_spectral(float(i) / n_clusters)
    ax1.fill_betweenx(np.arange(y_lower, y_upper),
                      0, ith_cluster_silhouette_values,
                      facecolor=color, edgecolor=color, alpha=0.7)

    # Label the silhouette plots with their cluster numbers at the middle
    ax1.text(-0.05, y_lower + 0.5 * size_cluster_i, str(i))

    # Compute the new y_lower for next plot
    y_lower = y_upper + 10 # 10 for the 0 samples

ax1.set_title("The silhouette plot for the various clusters.")
ax1.set_xlabel("The silhouette coefficient values")
ax1.set_ylabel("Cluster label")

# The vertical line for average silhouette score of all the values
ax1.axvline(x=silhouette_avg, color="red", linestyle="--")

ax1.set_yticks([]) # Clear the yaxis labels / ticks
ax1.set_xticks([-0.1, 0, 0.2, 0.4, 0.6, 0.8, 1])

# 2nd Plot showing the actual clusters formed
colors = cm.nipy_spectral(cluster_labels.astype(float) / n_clusters)
ax2.scatter(X[:, 0], X[:, 1], marker='.', s=30, lw=0, alpha=0.7,
            c=colors, edgecolor='k')


```

```

# Labeling the clusters
centers = clusterer.cluster_centers_
# Draw white circles at cluster centers
ax2.scatter(centers[:, 0], centers[:, 1], marker='o',
            c="white", alpha=1, s=200, edgecolor='k')

for i, c in enumerate(centers):
    ax2.scatter(c[0], c[1], marker='$%d$' % i, alpha=1,
                s=50, edgecolor='k')

ax2.set_title("The visualization of the clustered data.")
ax2.set_xlabel("Feature space for the 1st feature")
ax2.set_ylabel("Feature space for the 2nd feature")

plt.suptitle(("Silhouette analysis for KMeans clustering on sample data "
             "with n_clusters = %d" % n_clusters),
             fontsize=14, fontweight='bold')

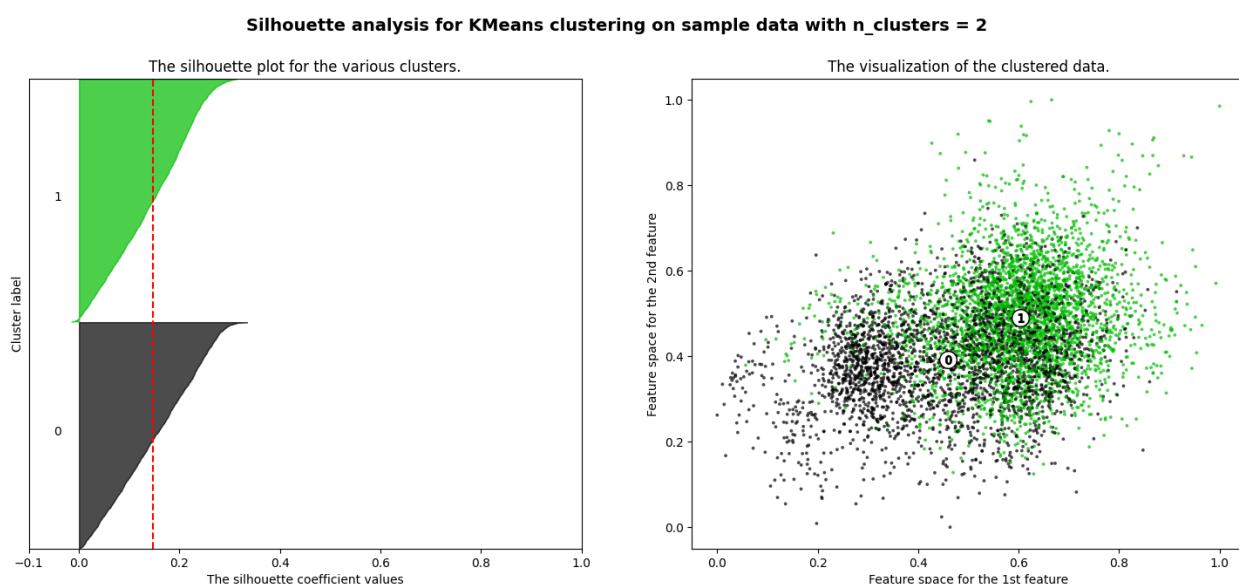

```

نمودارهای مربوطه در شکل‌های ۸۱-۸۴ آورده شده‌اند. در اینجا ذکر نکاتی جهت آشنایی بیشتر با این آنالیز الزامی است:

اول آنکه خط نقطه چین عمودی قرمز رنگ در نمودارهای silhouette plot بیانگر پارامتر average silhouette score می‌باشد.

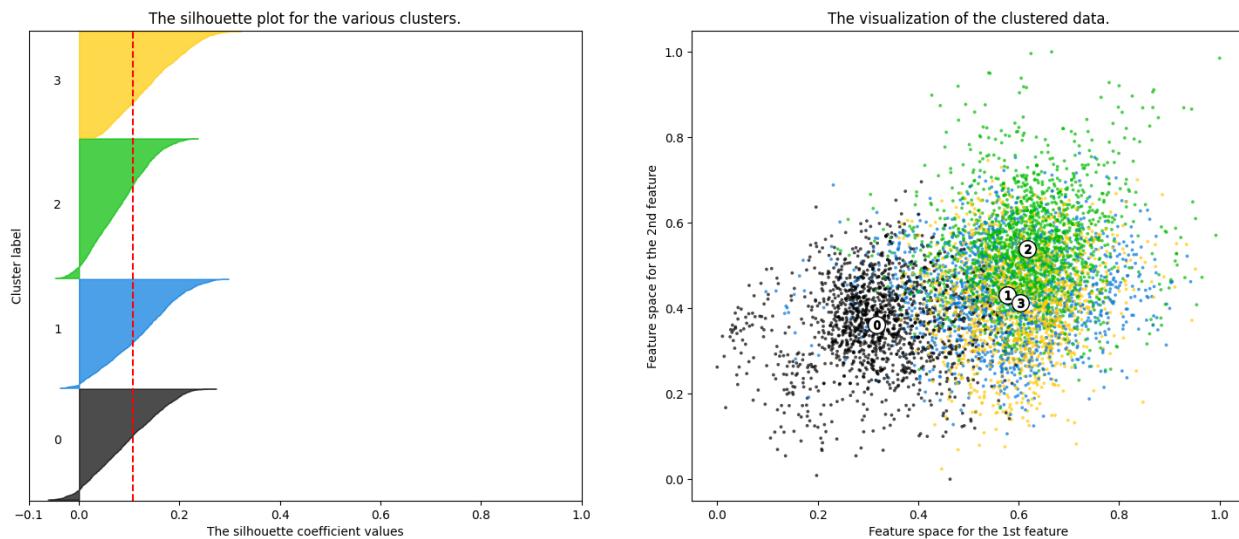
دوم آنکه برای انتخاب تعداد خوش مناسب باید این خط نقطه چین را در نظر داشته باشیم و آن تعداد خوش‌های را انتخاب کنیم که پهنه‌ای نمودارهای عمودی روی این نقطه خط چین همگن‌تر باشد. (به آن معنا که پهنه‌ای باندهایی که با رنگ مختلف رسم شده اند زیاد تغییر نکند.)

با توجه به این توضیحات به سراغ نمودارها می‌رویم:



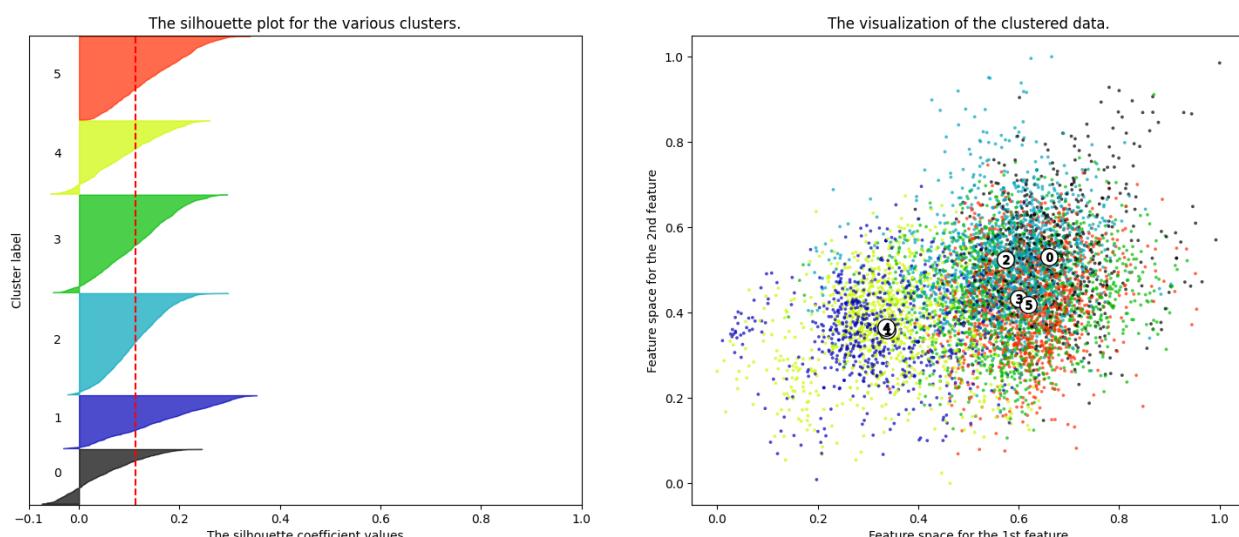
شکل ۸۱. آنالیز silhouette در الگوریتم K-means با تعداد کلاستر ۲

#### Silhouette analysis for KMeans clustering on sample data with n\_clusters = 4

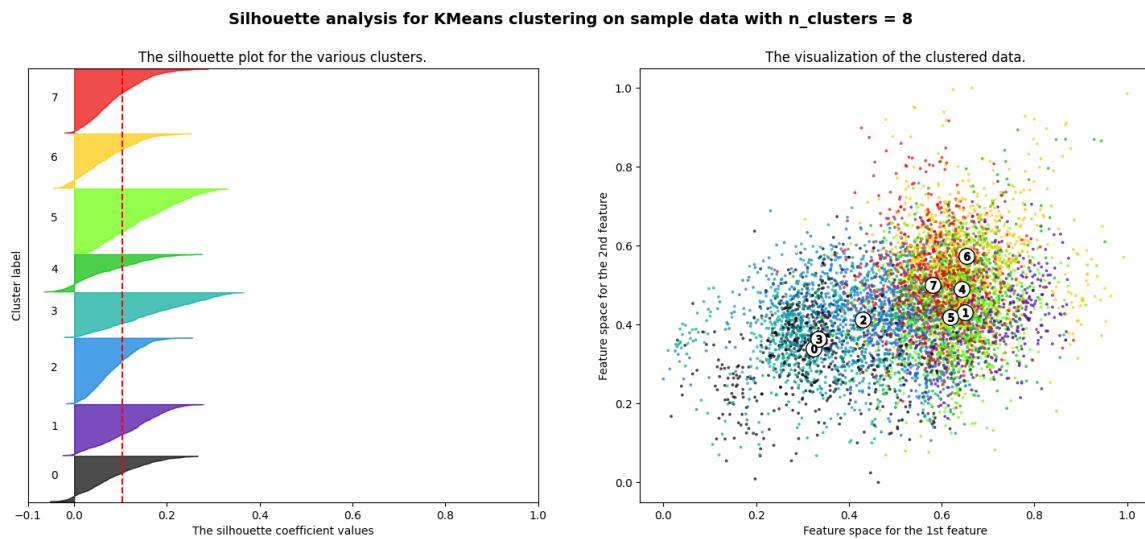


شكل ٨٢. آنالیز silhouette در الگوریتم K-means با تعداد کلاستر ٤

#### Silhouette analysis for KMeans clustering on sample data with n\_clusters = 6



شكل ٨٣. آنالیز silhouette در الگوریتم K-means با تعداد کلاستر ٦



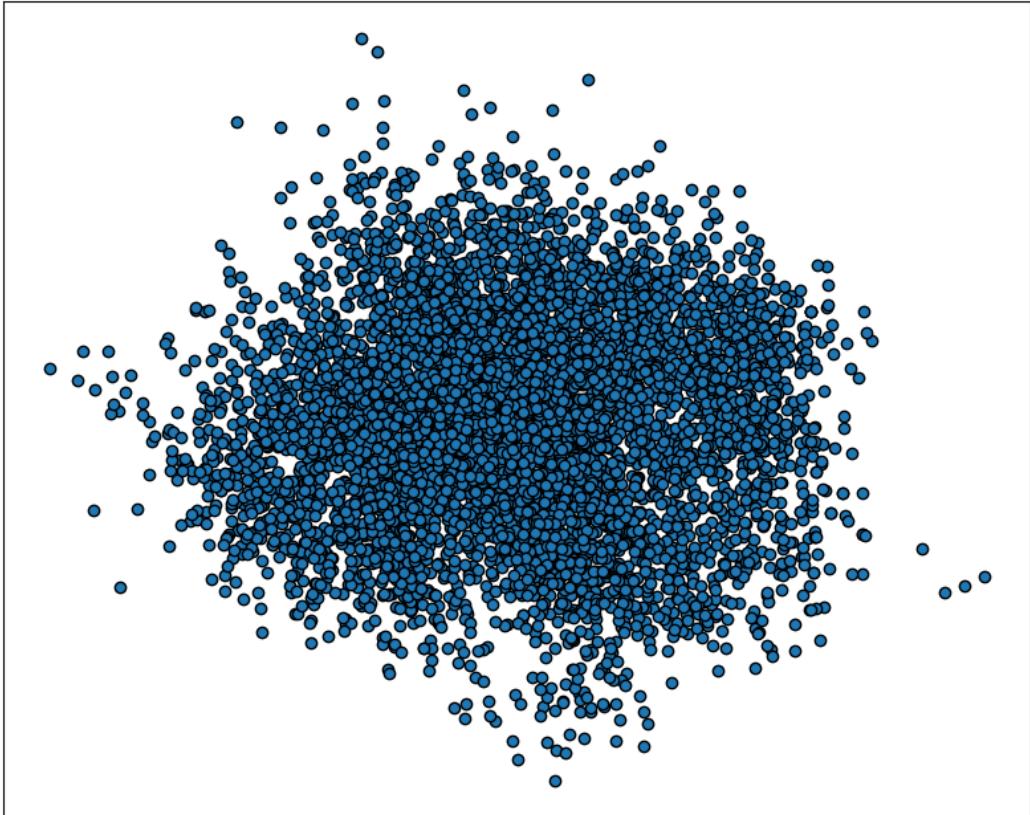
شکل ۸۴. آنالیز silhouette در الگوریتم K-means با تعداد کلاستر ۸

با توضیحات گفته شده، از شکل های ۸۱-۸۴ استنتاج می کنیم که هر چه تعداد خوشه ها کمتر باشد (به طور مثال تعداد ۲ خوشه) عمل clustering بهتر انجام شده است.

اکنون برای آن که درک بیشتری از توزیع دادگان موجود و خوشه بندی انجام شده بر روی آنها به دست آوریم، با بهره گیری از الگوریتم PCA ابتدا دیتاست را بر روی دو بعد ترسیم کرده و سپس نمودار خوشه بندی انجام شده بعلاوه لیبل واقعی داده ها را رسم می کنیم.

```
[13] # Top 2 pca components
pca = PCA(n_components=2)
principalComponents = pca.fit_transform(X)
principalDF = pd.DataFrame(data = principalComponents , columns = ['pc1','pc2'])

▶ fig, ax = plt.subplots(figsize=(10,8))
ax.axis('on')
plt.scatter(principalDF['pc1'].T, principalDF['pc2'].T, edgecolor = 'k')
frame = plt.gca()
frame.axes.get_xaxis().set_visible(False)
frame.axes.get_yaxis().set_visible(False)
```



شکل .۸۵ در دو بعد با استفاده از Scatter plot PCA

خوشبندی انجام شده با استفاده از تابعی که به نام `plot_cluster` تعریف کردهایم به صورت زیر خواهد بود:

```
[15] def plot_clusters(data, label, algorithm, args, kwds):
    start_time = time.time()
    labels = algorithm(*args, **kwds).fit_predict(data)
    end_time = time.time()
    palette = sns.color_palette('PuRd_r', np.unique(labels).max() + 1)
    colors = [palette[x] if x >= 0 else (0.70, 0.70, 0.70) for x in labels]
    fig, [ax1,ax2] = plt.subplots(1,2,figsize=(8,4))
    ax1.axis('on')
    ax2.axis('on')
    ax1.scatter(data[data.columns[0]].T, data[data.columns[1]].T, c=colors)
    ax2.scatter(data[data.columns[0]].T, data[data.columns[1]].T,c =label )
    frame = plt.gca()
    frame.axes.get_xaxis().set_visible(False)
    frame.axes.get_yaxis().set_visible(False)
    ax1.set_title('Clusters found by {}'.format(str(algorithm.__name__)), fontsize=8)
    ax2.set_title('data in real labels', fontsize=8)
    # plt.text(-0.5, 0.7, 'Clustering took {:.2f} s'.format(end_time - start_time), fontsize=14)
```

▶ number\_of\_clusters=[2,4,6,8]
 for nc in number\_of\_clusters:
 plot\_clusters(principalDf,y, cluster.KMeans, (), {'n\_clusters':nc})

نتایج شبیه‌سازی‌ها در شکل‌های ۸-۸۶ آورده شده است.



شکل ۸۶. خوش‌های یافت شده از الگوریتم K-means با ۲ کلاستر و لیبل‌ها قبل از خوش‌بندی



شکل ۸۷. خوش‌های یافت شده از الگوریتم K-means با ۴ کلاستر و لیبل‌ها قبل از خوش‌بندی



شکل ۸۸. خوشه‌های یافت شده از الگوریتم K-means با ۶ کلاستر و لیبل‌ها قبل از خوشه‌بندی



شکل ۸۹. خوشه‌های یافت شده از الگوریتم K-means با ۸ کلاستر و لیبل‌ها قبل از خوشه‌بندی

از شکل‌ها مشخص است که خوشه‌بندی انجام شده است.

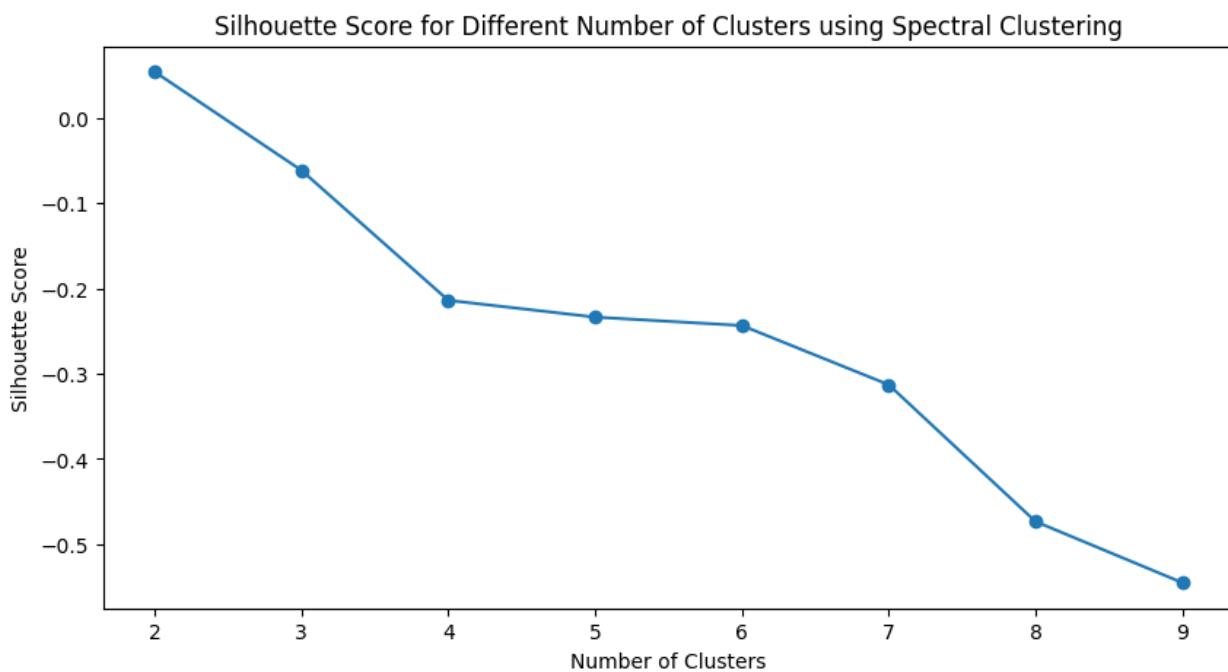
### Spectral Clustering-۳-۵

خوشه بندی طیفی را می توان به عنوان خوشه بندی نموداری در نظر گرفت. برای داده های مکانی می توان به القای یک نمودار بر اساس فواصل بین نقاط (به طور بالقوه یک نمودار kNN یا حتی یک نمودار متراکم) فکر کرد. از آنجا، خوشه بندی طیفی به بردارهای ویژه لاپلاسین گراف نگاه می کند تا تلاش کند یک جاسازی خوب (بعد پایین) نموداری در فضای اقلیدسی پیدا کند. این اساساً نوعی یادگیری چندگانه است، یافتن دگرگونی فضای اصلی به گونه ای است که فاصله های چندگانه را برای چند منیفولی که داده ها فرض می شود در نظر می گیرد، بهتر نشان دهد.

اکنون به سراغ کد پیاده سازی شده برای این قسمت می رویم.

در اینجا دقیق شود که تمامی مراحل آماده سازی دادگان و انجام عملیات نرمالیزه کردن مطابق قسمتهای قبلی بوده است.

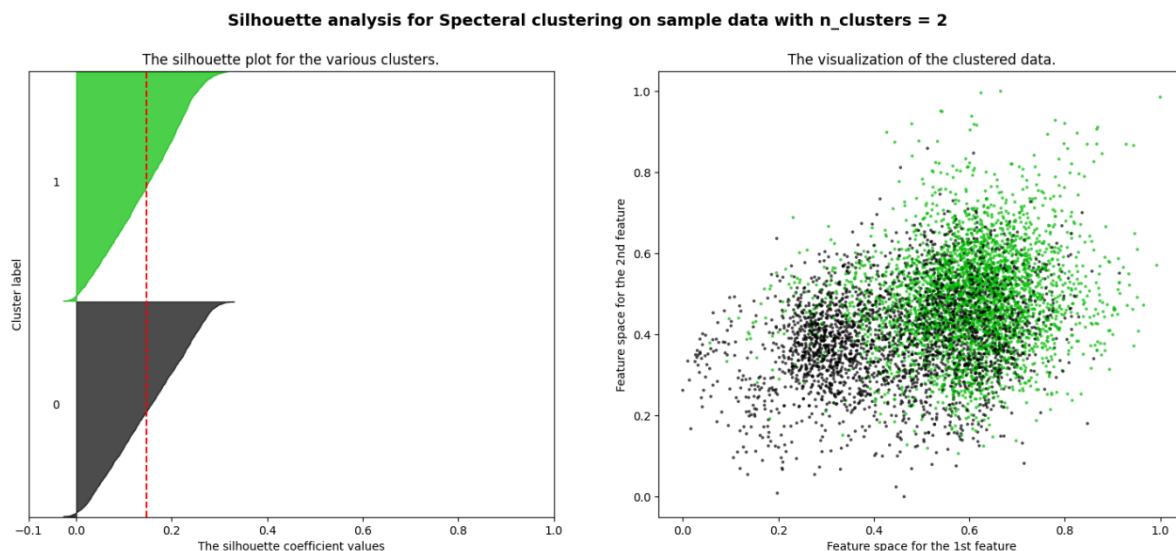
حال قبل از آنکه به اجرای کامل آنالیز silhouette score بپردازیم، نمودار silhouette score را به ازای تعداد خوشه های مختلف در شکل ۹۰ رسم می کنیم. (کد نوشته شده عیناً مشابه حالت k-means بوده و صرفاً از یک خوشه بند دیگر یعنی spectral clustering استفاده شده است).



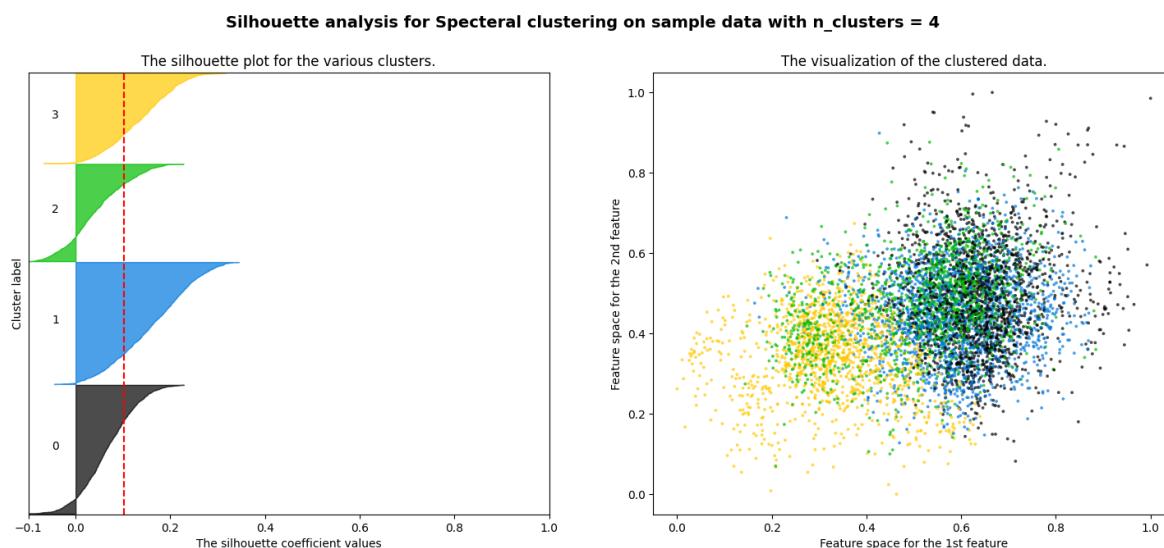
شکل ۹۰. محاسبه مقادیر silhouette score برای تعداد کلاسترها با روش spectral clustering

در اینجا نیز ملاحظه می‌شود مناسبترین تعداد خوش‌ها مطابق قبل برابر ۲ خوش است. (چون مقدار آن عددی مثبت می‌باشد)

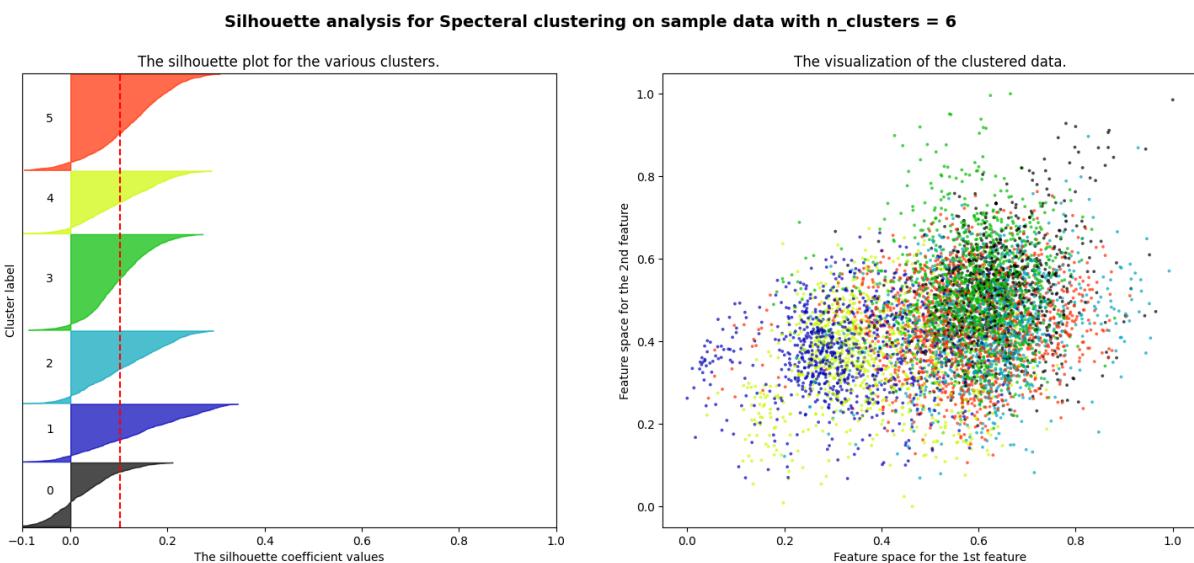
حال برای اجرای K-means که نوشته شده مشابه کد بخش مربوط به الگوریتم silhouette analysis می‌باشد، با این تفاوت که الگوریتم خوش‌بندی را به Spectral Clustering تغییر داده ایم؛ نتایج حاصل شده در شکل‌های ۹۱-۹۴ آورده شده است.



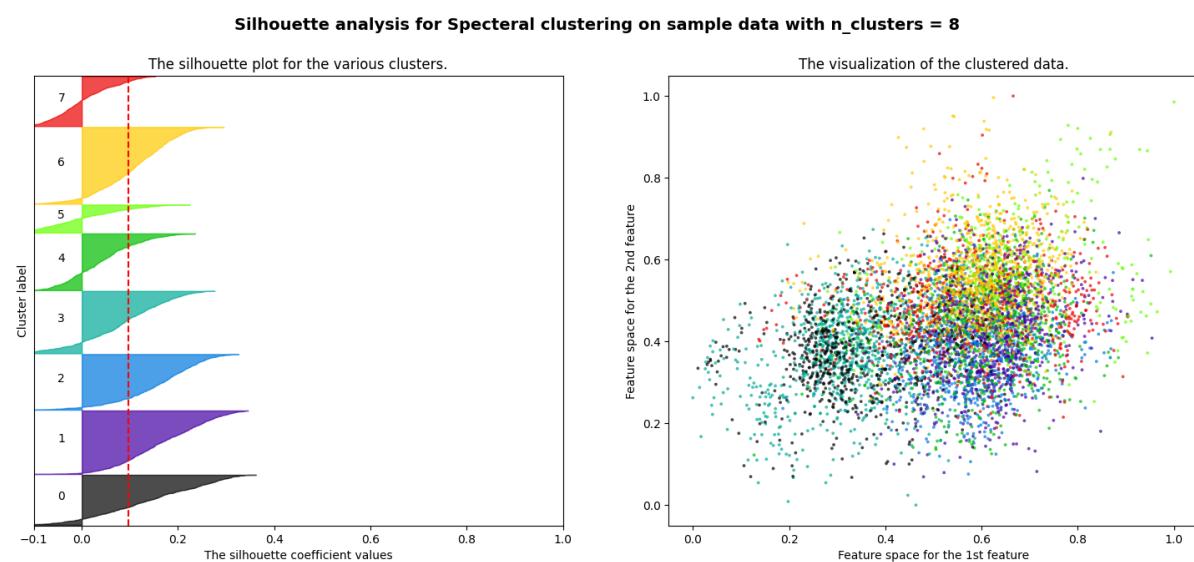
شکل ۹۱. آنالیز silhouette در روش spectral clustering با تعداد کلاستر ۲



شکل ۹۲. آنالیز silhouette در روش spectral clustering با تعداد کلاستر ۴



شکل ۹۳. آنالیز silhouette در روش specterl clustering با تعداد کلاستر ۶

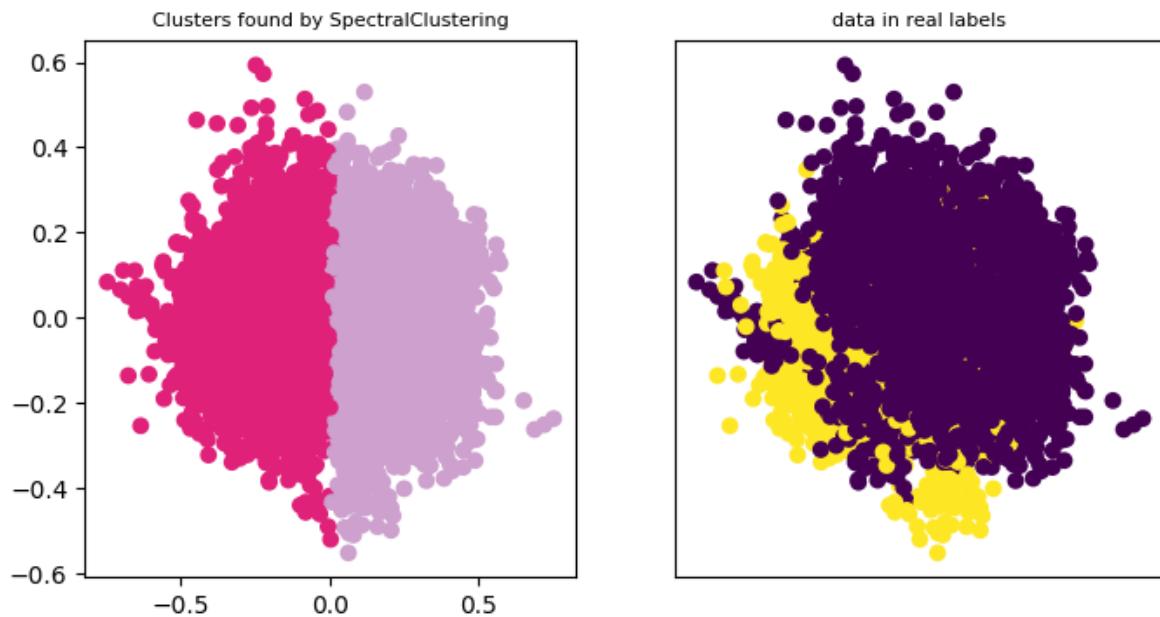


شکل ۹۴. آنالیز silhouette در روش specterl clustering با تعداد کلاستر ۸

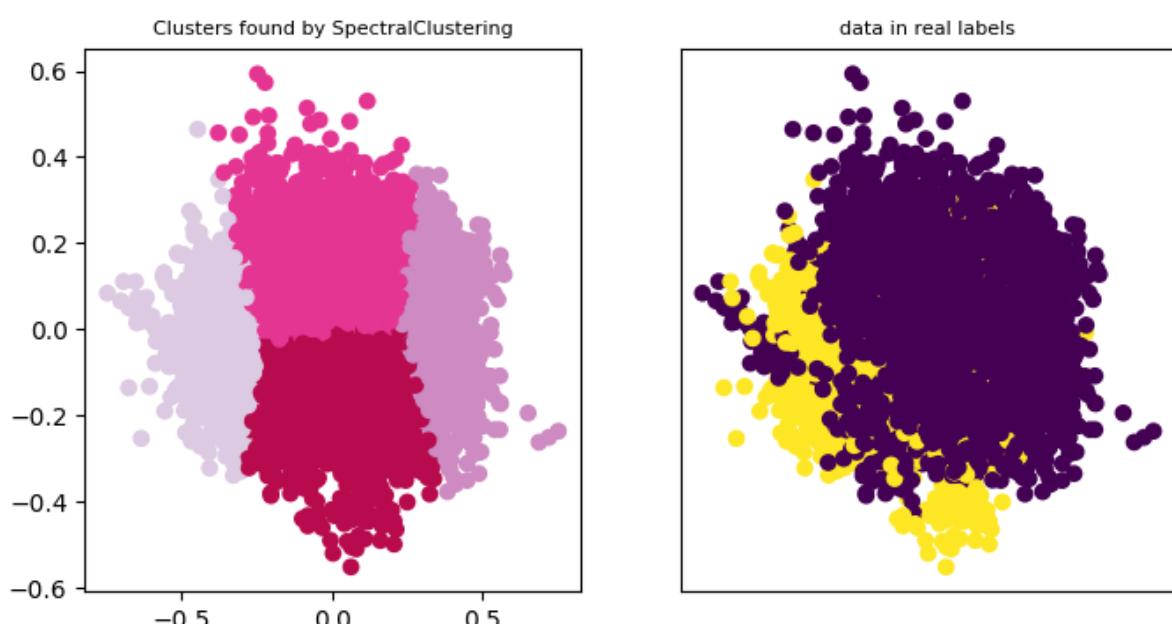
همانطور که ملاحظه می شود در اینجا نیز مطابق نتایج به دست آمده در بخش K-means تعداد مناسب خوشه ها برابر ۲ است. (علت آن که در خط عمودی رسم شده که بیانگر average silhouette score می باشد، تنها دو برای حالت دو خوشه، پهنه ای نمودارها همگن می باشد.)

بنابراین نتایج آنالیز silhouette برای روش های Spectral Clustering و K-means با هم همخوانی دارد.

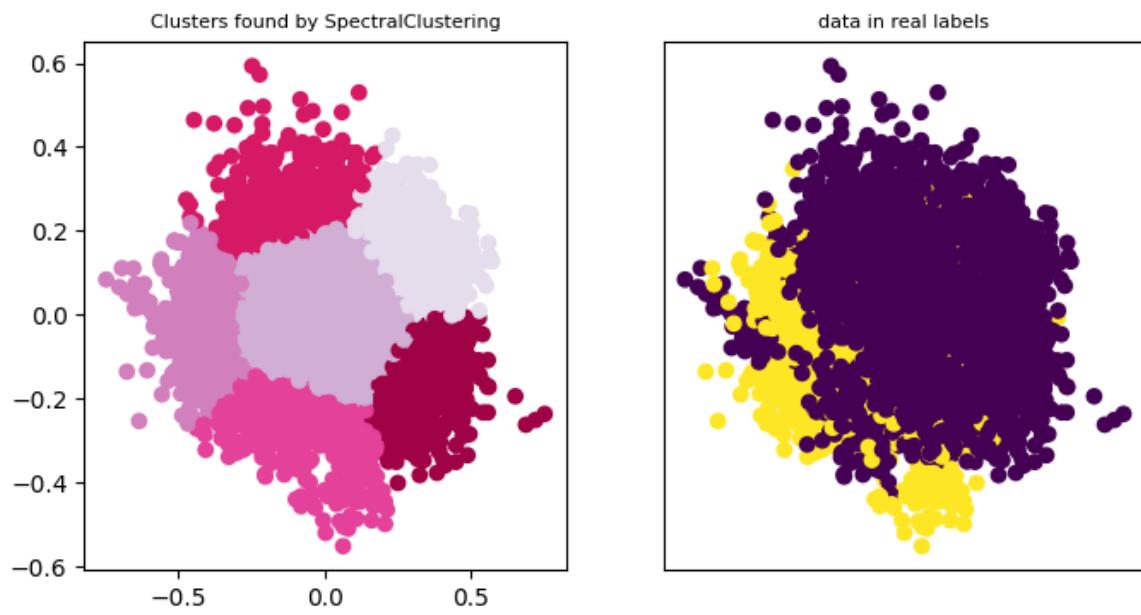
در پایان مجددا مشابه قبل نمودار خوشه‌های به دست آمده و همچنین نمودار واقعی دادگان به همراه لیبل آن‌ها را در دو بعد (کاهش بعد یافته به وسیله PCA) در شکل‌های ۹۵-۹۸ ترسیم می‌کنیم.



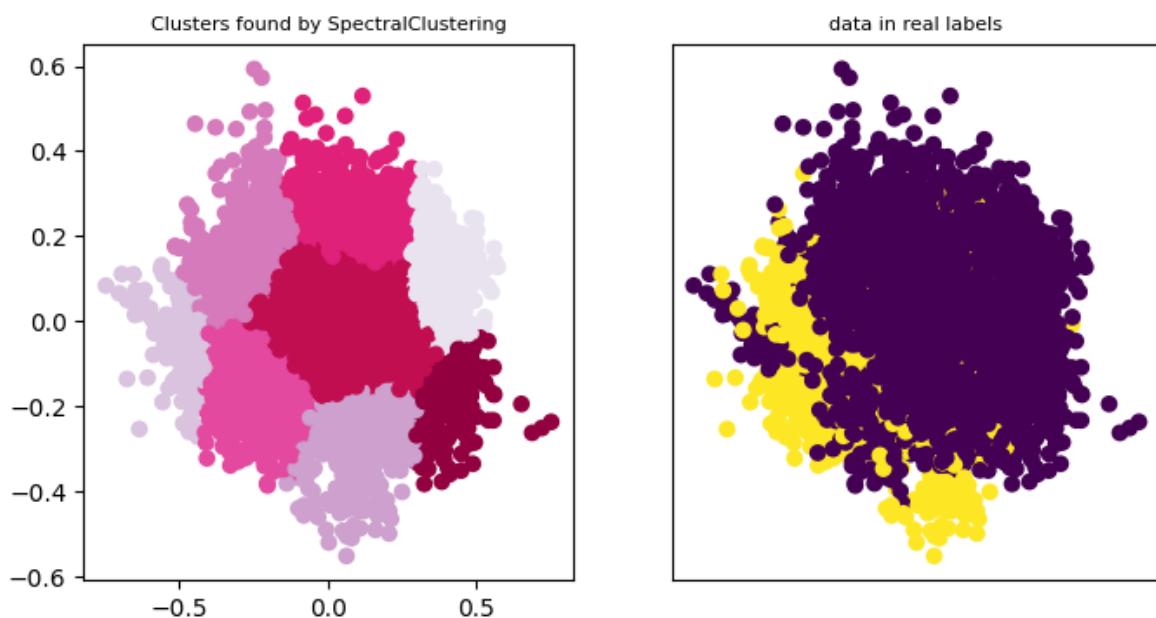
شکل ۹۵. خوشه‌های یافت شده از روش Spectral clustering با ۲ کلاستر و لیبل‌ها قبل از خوشبندی با کاهش بعد



شکل ۹۶. خوشه‌های یافت شده از روش Spectral clustering با ۴ کلاستر و لیبل‌ها قبل از خوشبندی با کاهش بعد



شکل ۹۷. خوشه‌های یافته شده از روش Spectral clustering با ۶ کلاستر و لیبل‌ها قبل از خوشه‌بندی با کاهش بعد



شکل ۹۸. خوشه‌های یافته شده از روش Spectral clustering با ۸ کلاستر و لیبل‌ها قبل از خوشه‌بندی با کاهش بعد

ملاحظه می‌شود که خوشه بندی با این روش تا حد زیادی با خوشه بندی با روش K-means مطابقت دارد و به درستی پیاده‌سازی صورت گرفته است.

### ۳-۶- نتیجه‌گیری و جمع‌بندی

با توجه به نوع داده‌ها و اینکه داده‌های صوتی به سادگی جدایی پذیر نیست، دقت پایین‌تر مدل‌ها توجیه پذیر می‌باشد. شایان ذکر است برای داده‌های صوتی بهتر است از مدل‌های حافظه دار مانند LSTM استفاده شود. همچنین عمل نرمالیزه کردن تنها زمانی مورد نیاز است که ویژگی‌های مدل‌های یادگیری ماشین دارای محدوده‌های متفاوتی باشند. برای داده‌های ما با توجه به اینکه جنس ویژگی‌ها مختلف بوده، و همچنین در رنج‌های مختلف قرار دارند لازم است که حتماً نرمالیزیشن صورت بگیرد. نرمال کردن داده‌ها در اکثر موارد باعث بهتر شدن نتیجه نهایی همه‌ی مدل‌ها می‌شود.

نکته دیگری که لازم است مورد توجه قرار بگیرد نویزی است که به دادگان آگشته شده: همانطور که ملاحظه کردیم بهترین خوش‌بندی با استفاده از ۲ کلاستر انجام شده است و در نمودارهای پراکندگی رسم شده نیز تا حد زیادی درهم تنیدگی نقاط قابل مشاهده بود. به همین علت کار تفکیک خوش‌های از همدیگر بسیار سخت بوده است.

## ۴- پیاده‌سازی ASR (بخش امتیازی)

### ۴-۱- مقدمه

امروزه دو رویکرد اصلی برای تشخیص خودکار گفتار یا همان ASR وجود دارد که یک رویکرد ترکیبی مرسوم و یک رویکرد یادگیری عمیق انتها به انتهای می‌باشد. در رابطه با این موارد در پیش‌گزارش به صورت مفصل توضیحاتی ارائه شد. در این قسمت ASR را به روش انتها به انتهای برای زبان فارسی انجام خواهیم داد. آموزش مدل‌های یادگیری عمیق انتها به انتهای، آسان‌تر است و به نیروی انسانی کمتری نسبت به رویکرد مرسوم و قدیمی نیاز دارد. همچنین از مدل‌های سنتی که امروزه استفاده می‌شوند نیز دقیق‌تر هستند. تحقیقاتی در جستجوی راههایی برای بهبود مداوم این مدل‌ها همچنان در جریان می‌باشد. در واقع ما شاهد خواهیم بود که مدل‌های یادگیری عمیق در چند سال آینده به دقتی در سطح انسانی می‌رسند.

### ۴-۲- پیش‌پردازش (Preprocessing)

کد ارائه شده در این قسمت، بخش پیش‌پردازش را برای مجموعه داده‌های در اختیار برای ASR با برای ما انجام خواهد داد. مراحل مختلفی که در این بخش انجام شده است را به اختصار توضیح خواهیم داد. ابتدای کار کتابخانه‌های لازم برای انجام این بخش را فراخوانی می‌کنیم. این کتابخانه‌ها برای مدیریت مجموعه داده‌ها، کار با مدل‌های ترانسفورمر و نرمال‌سازی متن فارسی بسیار مهم هستند که در کد ضمیمه شده قابل مشاهده می‌باشند.

در ادامه از تابع `reduce` برای ایجاد مجموعه‌ای از کاراکترهای منحصر به فرد موجود در ستون مجموعه داده‌ها استفاده خواهیم نمود. خروجی این کار شامل تعداد کاراکترهای منحصر به فرد و مجموعه واقعی کاراکترها است. اینکار می‌تواند برای بررسی تنوع کاراکترها در مجموعه داده مفید باشد، به خصوص قبل و بعد از هر مرحله پیش‌پردازش داده‌ها.

سپس از کتابخانه `hazm` برای نرمال‌سازی متن در ستون مجموعه داده‌ها استفاده نمودیم. نرمال‌سازی متن یک مرحله پیش‌پردازش رایج در تسک‌های پردازش زبان طبیعی برای اطمینان از سازگاری و بهبود عملکرد مدل است. مواردی که با استفاده از این کتابخانه انجام شد به شرح جدول ۲ است:

جدول ۲. موارد استفاده شده در نرمال‌سازی توسط کتابخانه hazm

نام	توضیحات
correct_spacing	فاصله‌گذاری‌ها را در متن، نشانه‌های سجاموندی و پیشوندها و پسوندها اصلاح می‌کند.
remove_diacritics	اعراب حروف را حذف می‌کند.
remove_specials_chars	برخی از کاراکترها و نشانه‌های خاص را که کاربردی در پردازش متن ندارند حذف می‌کند.
decrease_repeated_chars	تکرارهای بیش از ۲ بار را به ۲ بار کاهش می‌دهد. مثلاً «سلاممم» را به «سلامم» تبدیل می‌کند.
persian_numbers	ارقام انگلیسی را با فارسی جایگزین می‌کند.
unicodes_replacement	برخی از کاراکترهای یونیکد را با معادل نرمال‌شده آن جایگزین می‌کند.
seperate_mi	پیشوند «می» و «نمی» را در افعال جدا می‌کند.

پس از عملیات فوق، کاراکترهایی که صدایی ندارند، حذف گردیدند. علاوه بر این موارد، برخی علائم نیز با معادل آن‌ها جایگزین شد. به طور مثال:٪ با «درصد»

#### ۴-۳-آموزش (Training)

در بخش Training با توجه به اینکه فرایند آموزش بسیار طولانی بود (بیشتر از ۴ ساعت با GPU) برخی موارد از کلاس TrainingArguments را شخصی سازی کردیم بطوریکه به مشکل out of memory برنخوریم:

**per\_device\_train\_batch\_size**: افزایش اندازه دسته سرعت آموزش را افزایش میداد به مشکل محدودیت حافظه برخورد میکردیم.

**gradient\_accumulation\_steps**: کاهش این مقدار وزن بهروزرسانی مدل را بیشتر می‌کند. ولی گام‌های بسیار کم ممکن است منجر به آموزش ناپایدار شود، مخصوصاً در نرخ یادگیری بالا.

**num\_train\_epochs**: تعداد دوره‌ها را به ۳ کاهش دادیم چرا که در همین محدوده مقدار خطأ همگرا میشد و مشکلی از جهت underfitting پیدا نمیکردیم.

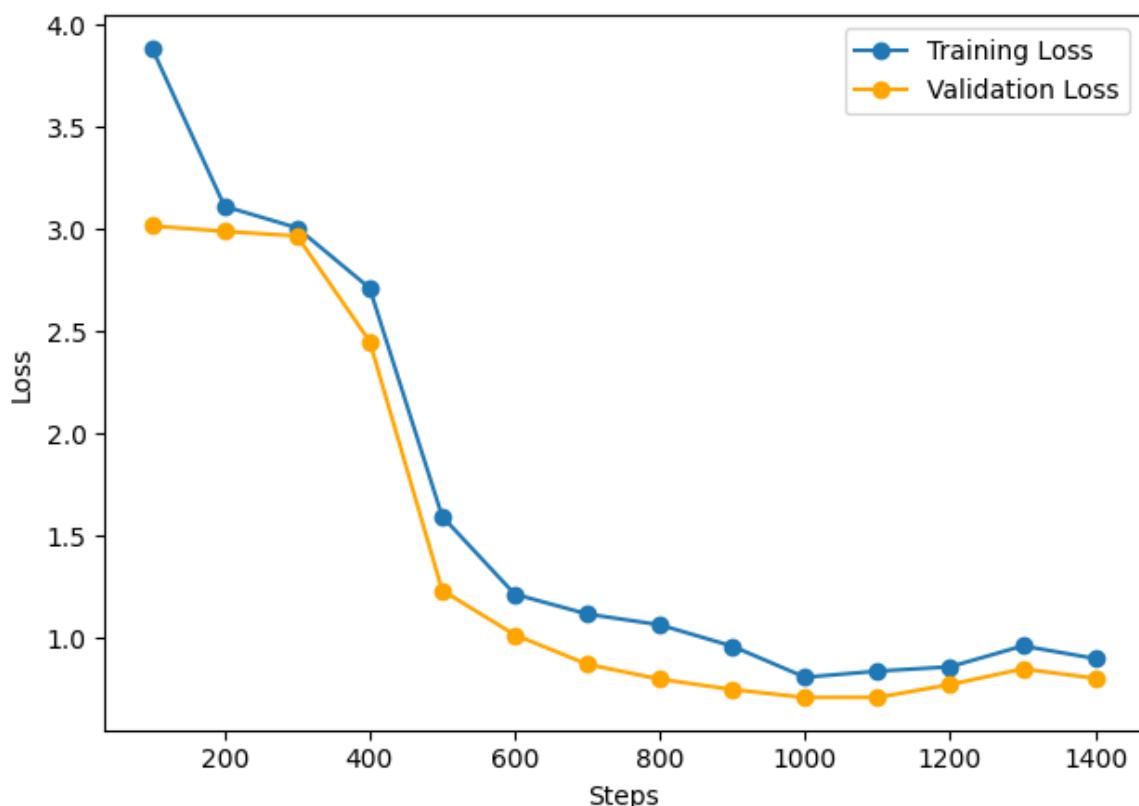
**learning\_rate**: نرخ یادگیری را از  $10^{-4} \times 3$  به  $10^{-4} \times 4$  افزایش دادیم بطوریکه باعث بی‌ثباتی در طول آموزش نشود.

برای کاهش دفعات ذخیره مدل، `save_steps` را افزایش دادیم. `save_total_limit` را روی عدد کمتری تنظیم کردیم تا نقاط بازرسی کمتری حفظ شود. `logging_steps` را افزایش دادیم تا فرکانس ثبت کاهش یابد، که می‌تواند کمی سرعت آموزش را افزایش دهد.

همچنین پارامتر `logging_dir` را برابر `./logs` کردیم تا بتوانیم با استفاده از داده‌های ذخیره شده، تغییرات خطای طول آموزش را رسم کنیم.

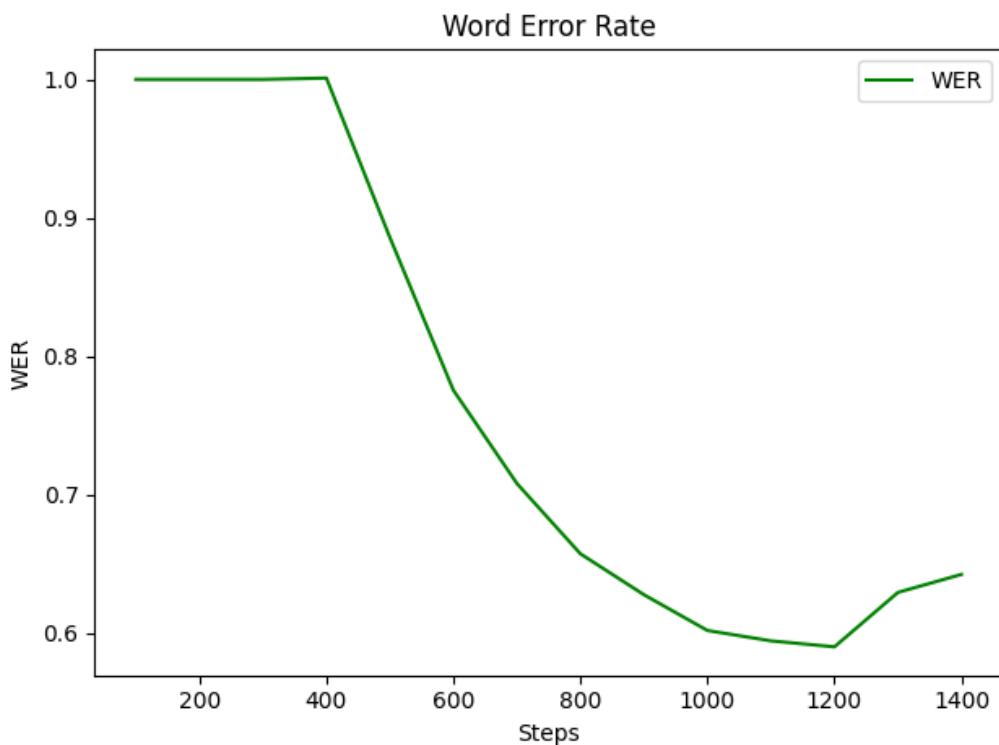
#### ۴-۴-۴- ارزیابی (Evaluation)

با مفروضاتی که انجام دادیم، مدل را آموزش دادیم. شکل ۹۹، نمودار تغییرات خطای طول آموزش را نشان می‌دهد.



شکل ۹۹. نمودار تغییرات خطای طول آموزش

شکل ۱۰۰ نیز نمودار تغییرات WER در طول آموزش را نشان می‌دهد.

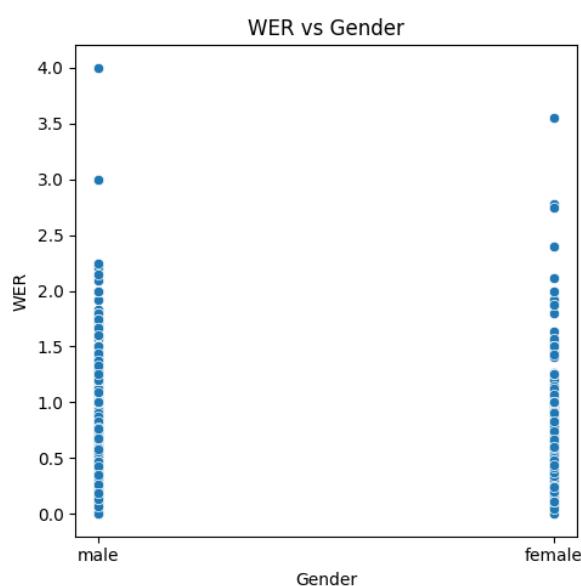


شکل ۱۰۰. نمودار تغییرات WER در طول آموزش

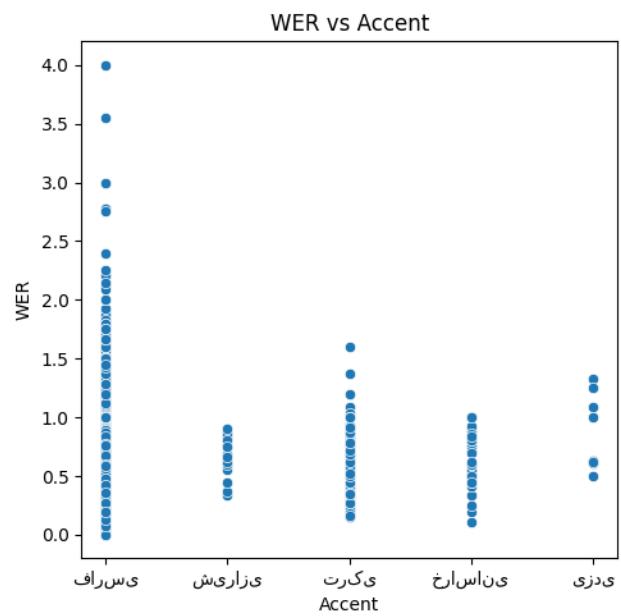
ملحوظه می‌شود که با افزایش گام‌های آموزشی، خطای آموزش و ارزیابی نیز کاهش پیدا می‌کند و آموزش به درستی اتفاق افتاده است.

#### ۴-۵-نتیجه‌گیری

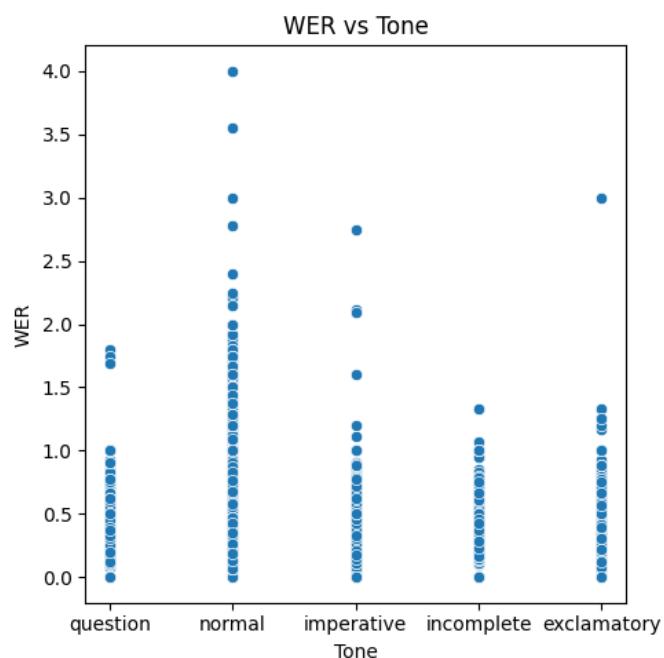
شکل‌های ۱۰۱ تا ۱۰۳، نمودارهای پراکنده‌ی WER براساس جنسیت، لهجه و لحن را نشان می‌دهند.



شکل ۱۰۱. نمودار پراکنده‌ی WER بر اساس جنسیت

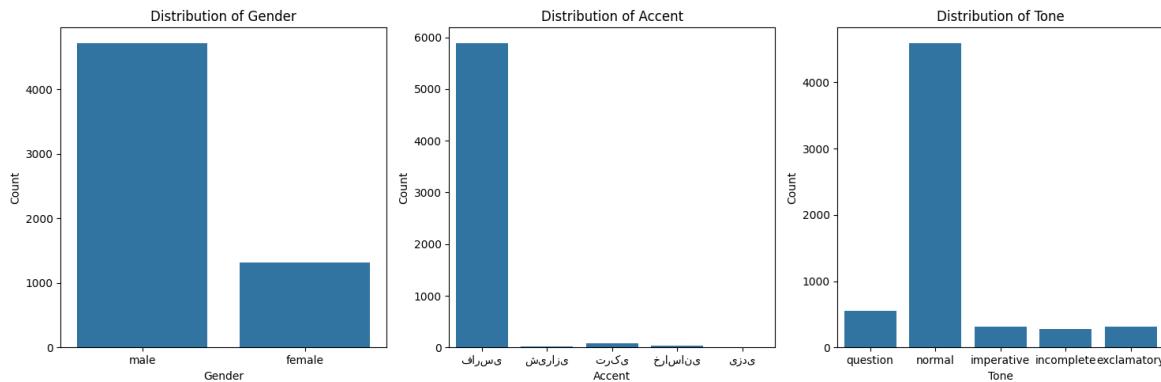


شکل ۱۰۲. نمودار پراکندگی WER بر اساس لهجه



شکل ۱۰۳. نمودار پراکندگی WER بر اساس لحن

شکل ۱۰۴ نیز نمودار توزیه ویژگی‌های جسمیت، لهجه و لحن را نشان می‌دهد.



شکل ۱۰۴. نمودار توزیع ویژگی های جنسیت، لهجه و لحن

شکل ۱۰۵ نیز همبستگی ویژگی های جنسیت، لهجه و لحن با WER را نمایش می دهد.

```
Correlation between WER and Gender: -0.016, P-value: 0.223
Correlation between WER and Tone: 0.027, P-value: 0.035
Correlation between WER and Accent: -0.064, P-value: 0.000
```

شکل ۱۰۵. محاسبه همبستگی ویژگی های جنسیت، لهجه و لحن با WER

همبستگی جنسیت با خطای نسبت اینکه جمعیت گوینده های مرد بیشتر از زن بود، اما در کل خطای کمتری داشتن نسبت به زن ها داشتند که علت آن می تواند مواردی از قبیل حساسیت به تن صدا و... در نظر گرفت.

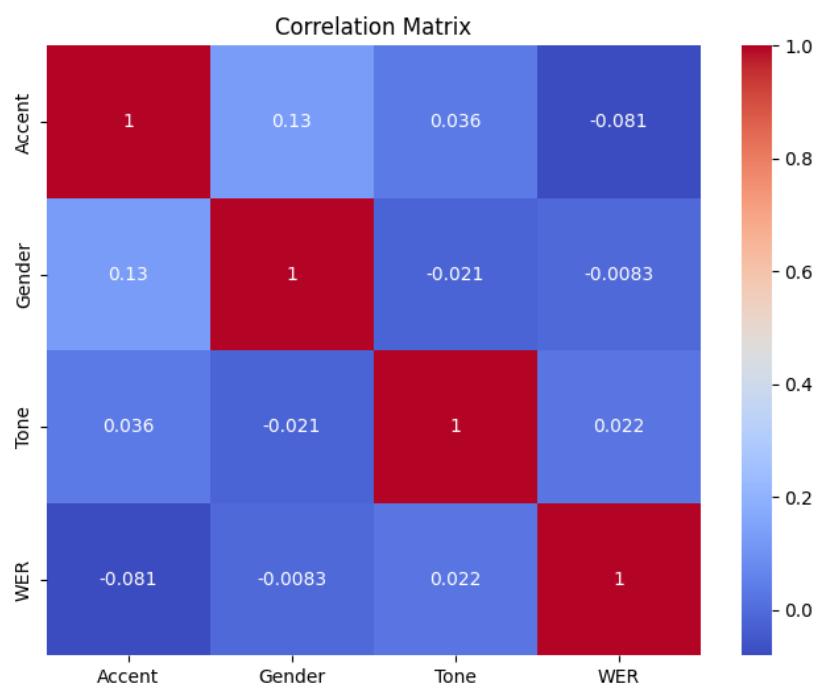
لهجه ها ممکن است بر دقت ASR تاثیرگذار باشند. سیستم هایی که بیشتر با یک لهجه خاص آموزش دیده اند، ممکن است در تشخیص گفتار با لهجه های دیگر کمتر دقیق باشند. این مسئله در مورد لهجه های منطقه ای، ملی یا حتی اجتماعی صدق می کند. به طور مثال بررسی شد که در زبان یزدی، ملاحظه شد که خطای آن بطور میانگین اندکی بیشتر بود.

گاهی اوقات، سیستم های ASR بهتر با صدای خاصی که در داده های آموزشی بیشتر موجود بوده اند، کار می کنند. اگر سیستم بیشتر با صدای مردان آموزش داده شده باشد، ممکن است در تشخیص گفتار زنان کمی ضعیفتر عمل کند و بالعکس که در دیتاستی که در اختیار داشتیم، تعداد داده های آموزش بیشتر با صدای مردان بود و این دلیلی برای آنکه خطای نمونه های زنان بیشتر می باشد است.

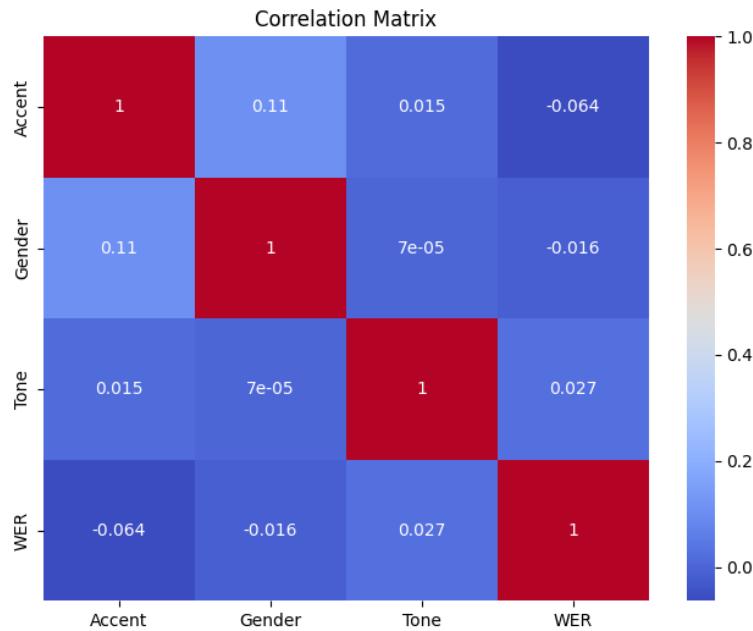
توضیحات داده شده در رابطه با لحن نیز صدق می کنند. اما این ویژگی ها در کل همبستگی خیلی کمی با خطای داشتند. همچنین با توجه به شکل ۱۰۴ و ملاحظه توزیع ها، دیده می شود که این توزیع ها بسیار ناهمگون می باشند و در هر مورد داده های یکی بسیار بیشتر از سایرین است. این مشکل عدم تعادل در دیتاهای بحث fair sample را مطرح می کند که می تواند در دقت های به دست آمده و آموزش مدل تاثیر

منفی بگذرد. همچنین منجر به حساسیت مدل نسبت به ویژگی‌هایی همچون جنسیت، لهجه و لحن می‌شود و با کوچکترین تغییر در هر مورد ممکن است نتایج بسیار فرق بکند. در واقع بالانس بودن داده‌ها و fair sample بودن اهمیت بسیاری در آموزش مدل‌ها و قدرت تعمیم‌پذیری آن‌ها خواهد داشت.

در ادامه نیز ماتریس همبستگی با متدهای pearson و spearman را در شکل‌های ۱۰۶ و ۱۰۷ نمایش دادیم. به خوبی همبستگی میان جنسیت، لحن، لهجه و WER را نمایش می‌دهد که جنسیت و لهجه ارتباط بیشتری با یکدیگر دارند.



شکل ۱۰۶. ماتریس همبستگی (متد spearman)



شکل ۷.۱۰۷. ماتریس همبستگی (متد pearson)

## جمع‌بندی و نتیجه‌گیری

در این گزارش ابتدا توضیحاتی پیرامون نحوه تمیز کردن داده‌ها و اطلاعات و استخراج ویژگی دادیم و سپس با انجام موارد گفته شده، ویژگی‌ها را به دست آوردیم تا بتوانیم در بخش‌های بعدی گزارش از آن‌ها استفاده کنیم. سپس مدل‌های مختلفی مانند Linear SVM, RBF SVM, Logistic Regression, MLP, Ensemble Learning و Naïve Bayes را توضیح مختصر داده و برای مسئله طبقه‌بندی به کار گرفتیم و در حالت‌های نرمال‌سازی شده داده‌ها و بدون نرمال‌سازی شده مدل‌ها را آموزش دادیم و عملکرد آن‌ها را بررسی نمودیم. همچنین با استفاده از روش‌های کاهش بعد مانند PCA و LDA، ابعاد ویژگی را کاهش داده و با فرضیات اشاره شده، مجدداً مدل‌ها را آموزش دادیم و در نهایت در جدول ۱ دقت‌های به دست آمده در هر روند آموزشی را آورده و مقایسه‌ای بین عملکرد مدل‌ها صورت دادیم. در بخش بعدی نیز سراغ مسئله خوشه‌بندی رفتیم و با استخراج ویژگی‌های موجود، الگوریتم‌های Spectral clustering و K-Means را با چند تعداد کلاستر مختلف اجرا نمودیم و نتایج آن را نیز بررسی کرده و در بخش مورد نظر آورده‌یم.

در انتهای بخش امتیازی پروژه یعنی ASR را انجام دادیم که شامل بخش پیش‌پردازش داده‌ها، آموزش و ارزیابی می‌باشد. توضیحات هر یک از این بخش‌ها در قسمت مربوط به خود آورده شده و روی آن‌ها بحث نمودیم و در انتهای حساسیت مدل آموزش دیده شده نسبت به ویژگی‌هایی همچون جنسیت و لهجه را ملاحظه نمودیم و همبستگی این موارد به انضمام لحن را در ماتریس همبستگی نمایش دادیم. تمامی کدهای زده شده، طبق محل‌های مورد نظر قرار گرفته شده در سامانه ایلرن بارگذاری شدند.