Gender Detection

Selen Akkaya s289332

January 2023

Abstract

In this paper, 'synthetic speaker embeddings that represent the acoustic characteristics of a spoken utterance' is analyzed and a gender classification task is applied by building commonly used machine learning algorithms. Moreover, the performances of applied machine learning models and the comparison of models are analyzed.

1 Introduction

1.1 Problem Overview

The data-set contains synthetic speaker embeddings which represent the acoustic characteristics of a spoken utterance. Each row corresponds to a different speaker and contains **12 features** followed by the gender label:
1: female,
0: male
The features do not have any particular interpretation. Speakers belong to four different age groups. The age information, however, is not available.
The **training set** consists of **3000** samples for each class, whereas the **test set** contains **2000** samples for each class.

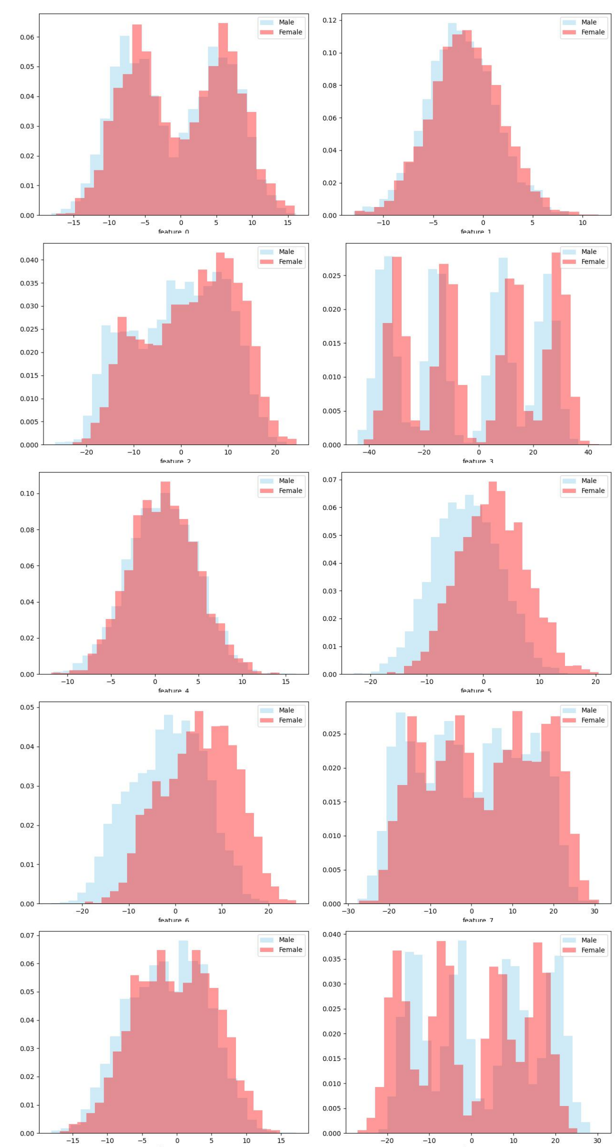
1.2 Exploratory Data Analysis

The 12 features are in a scale that have considerably similar means and variances, so it does not worth to apply Z-normalization which is basically centering every feature to its mean and scaling to unit variance xi = (xi-μ) / σ

μ : [-0.40439904, -1.98045219, 0.84747715, -2.37863374, 0.97348671, -0.72096827, 1.684338, 1.49200716, -0.8046595, 1.31572434, -0.07712583, 1.00468738]

σ : [7.09209235, 3.52880203, 9.8027367, 23.02600239, 3.85825232, 6.35299195, 8.5832784, 13.35106596, 5.71561775, 13.29758557, 10.69372272, 6.69376688]

Histograms of each 12 features (raw data) are shown below as Figure 1. It is obvious that raw features have approximated Gaussian distribution.



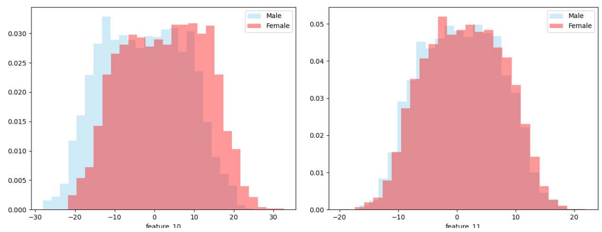
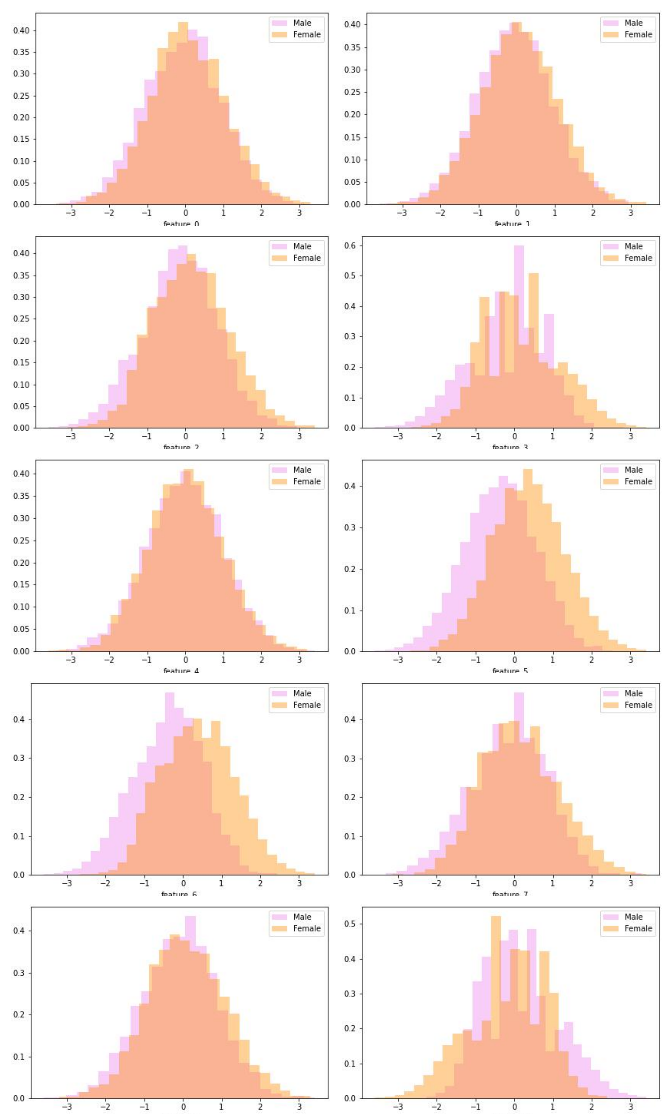


Figure 1: Raw Features

However, to approve this idea, histograms of Gaussianized features are plotted to demonstrate. Every 12 features with Gaussianization as it is shown in Figure 2.



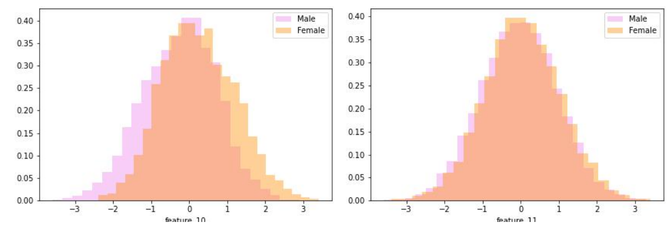


Figure 2: Gaussianized Features

The gaussianization did not improve the histograms. Using raw data is better in this case. Especially feature-3 and feature-9 show how gaussianization worsens the result compared to the raw data.

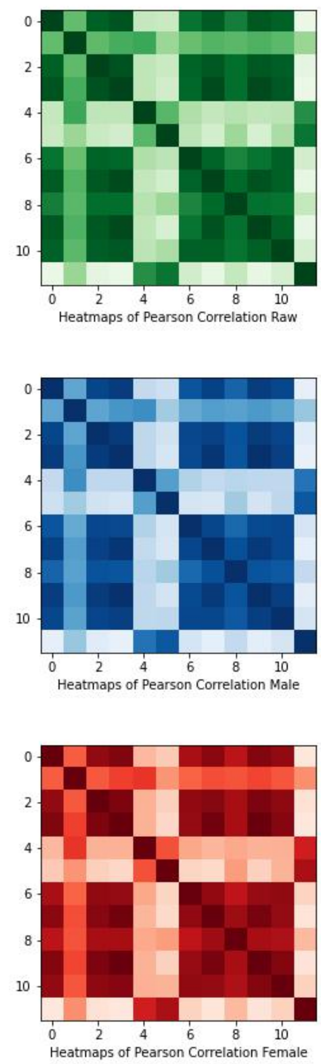


Figure 3: : Heatmap - Pearson Correlation

Pearson Correlation Heatmap shows that there are strongly corrolated features for instance, feature 3 is highly correlated to the 0, 2, 7 and 9. As an another example, feature 10 has a reasonable correlation with 0, 2, 3, 6, 7 and 9. Therefore, we can benefit of PCA to reduce dimension and map data to less correlated features.

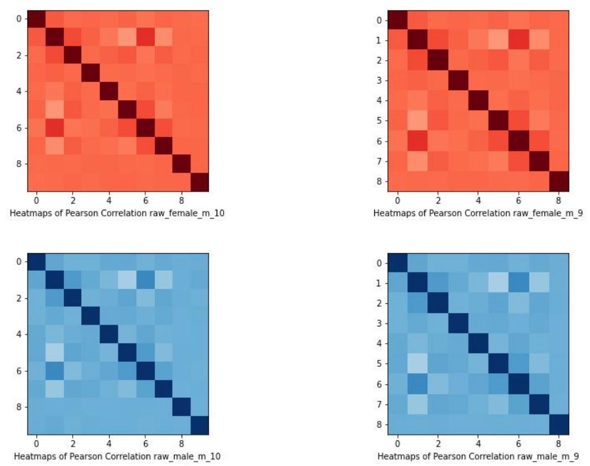


Figure 4: : Heatmap - Pearson Correlation with PCA m=10 and PCA m=9

2 Classification

In this report, the following machine learning models will be implemented and collected outputs will be compared:
****

- Generative models - Linear and Quadratic Classifiers

– Multivariate Gaussian Classifier (MVG)
– MVG + Diagonal Covariance
– MVG + Tied Covariance
– MVG + Diagonal Covariance with Tied Covariance

- Logistic Regression

– Quadratic Logistic Regression
– Prior Weighted Logistic Regression

- Support Vector Machine

– Linear SVM
– Quadratic SVM (polynomial kernel function with degree=2)
– SVM with Radial Basis kernel Function

- Gaussian Mixture Models

– Gaussian Mixture Models (GMM)
– GMM + Diagonal Covariance
– GMM + Tied Covariance
– GMM + Diagonal Covariance with Tied Covariance

For what concerns validation, it is necessary to make the following clarifications: • To understand which model is most promising, and to assess the effects of using PCA, we have employed K-Fold cross validation. In fact, all of the following results have been obtained with K-Fold Validation with K = 5. • Inside each cell of the following tables, we have reported the minDCF. We do not care about actDCF in this initial phase. • 'MinDCF' has been computed with Cfp and Cfn both equal to one, as we do not have any specific requirements regarding the miss-classification costs. In particular, we will consider
– a balanced case (our application):
(π, Cfn, Cfp) = (0.5, 1, 1)
– two unbalanced cases:
(π, Cfn, Cfp) = (0.1, 1, 1)
(π, Cfn, Cfp) = (0.9, 1, 1)
****

2.1 Multivariate Gaussian Classifiers

Samples of each class (male, female) can be modeled as samples of Multivariate Gaussian Classifiers (MVG) with class dependent mean and covariance matrices.
In particular, with the covariance matrices that are Full Covariances, Tied Covariance, Diagonal Covariances. These are generative models with Gaussian distributed data, given the class, as:

$$X|C = c \sim N(\mu_c, \Sigma_c)$$

Tied Multivariate Gaussian Classifiers assume that each class has its own mean, but the covariance matrix is the same for all classes.

$$X|C = c \sim N(\mu_c, \Sigma)$$

Naive Bayes version of MVG is simply a Gaussian Classifier where the covariance matrices are diagonal. We can adopt MVG by simply zeroing the out-of-diagonal elements of MVG solution.

It is obvious from previous histograms that features approximately have a gaussian distribution. Therefore, Generative Models should work well with our dataset.
Apart from that, it is expected to have poor performance from the Naive Bayes classifier since heatmaps show that correlation is significantly spreaded between the features.

The results that are obtained are below (for Gaussian Classifiers) for Raw features, Z-normed features and Gaussianized features. It is shown with different applications (ours has $= 0.5$) with π=0.5, π=0.1 and π=0.9:
no PCA, PCA-m=10 and PCA-m=10
K = 5

RAW Features, no PCA, K = 5,

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Full Covariance - Untied	0.048	0.126	0.123
Naive Bayes - Untied	0.565	0.818	0.848
Full Covariance - Tied	0.048	0.125	0.127
Naive Bayes - Tied	0.566	0.821	0.845

Z-Normalized Features, no PCA, K = 5,

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Full Covariance - Untied	0.048	0.126	0.123
Naive Bayes - Untied	0.565	0.818	0.848
Full Covariance - Tied	0.048	0.125	0.127
Naive Bayes - Tied	0.566	0.821	0.845

Gaussianized Features, no PCA, K = 5,

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Full Covariance - Untied	0.062	0.181	0.171
Naive Bayes - Untied	0.541	0.810	0.824
Full Covariance - Tied	0.060	0.180	0.167
Naive Bayes - Tied	0.538	0.804	0.816

RAW Features, PCA m = 10, K = 5,

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Full Covariance - Untied	0.047	0.140	0.120
Naive Bayes - Untied	0.067	0.173	0.161
Full Covariance - Tied	0.048	0.131	0.124
Naive Bayes - Tied	0.067	0.166	0.158

Z-Normalized Features, PCA m = 10, K = 5,

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Full Covariance - Untied	0.112	0.140	0.120
Naive Bayes - Untied	0.119	0.173	0.161
Full Covariance - Tied	0.111	0.131	0.124
Naive Bayes - Tied	0.118	0.166	0.158

Gaussianized Features, PCA m = 10, K = 5,

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Full Covariance - Untied	0.071	0.206	0.204
Naive Bayes - Untied	0.085	0.228	0.223
Full Covariance - Tied	0.071	0.199	0.206
Naive Bayes - Tied	0.083	0.227	0.225

RAW Features, PCA m = 9, K = 5,

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Full Covariance - Untied	0.046	0.137	0.121
Naive Bayes - Untied	0.067	0.171	0.159
Full Covariance - Tied	0.047	0.130	0.122
Naive Bayes - Tied	0.066	0.163	0.158

Z-Normalized Features, PCA m = 9, K = 5,

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Full Covariance - Untied	0.158	0.403	0.376
Naive Bayes - Untied	0.161	0.417	0.377
Full Covariance - Tied	0.155	0.398	0.367
Naive Bayes - Tied	0.161	0.415	0.376

In overall, full covariance matrices perform better than diagonal covariance matrices. This was expected because of highly correlated features.

Gaussianized Features, PCA m = 9, K = 5,

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Full Covariance - Untied	0.091	0.242	0.238
Naive Bayes - Untied	0.095	0.260	0.258
Full Covariance - Tied	0.090	0.236	0.233
Naive Bayes - Tied	0.096	0.260	0.261

PCA gives a good result even with m=9. It was expected since correlation is obvious between features (from heatmaps). Moreover, PCA improves the Naive Bayes performance. However it is not significant compared to full covariance models on RAW features. Apart from that, Z normalization features does not bring a remarkable result.

2.2 Logistic Regression