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.

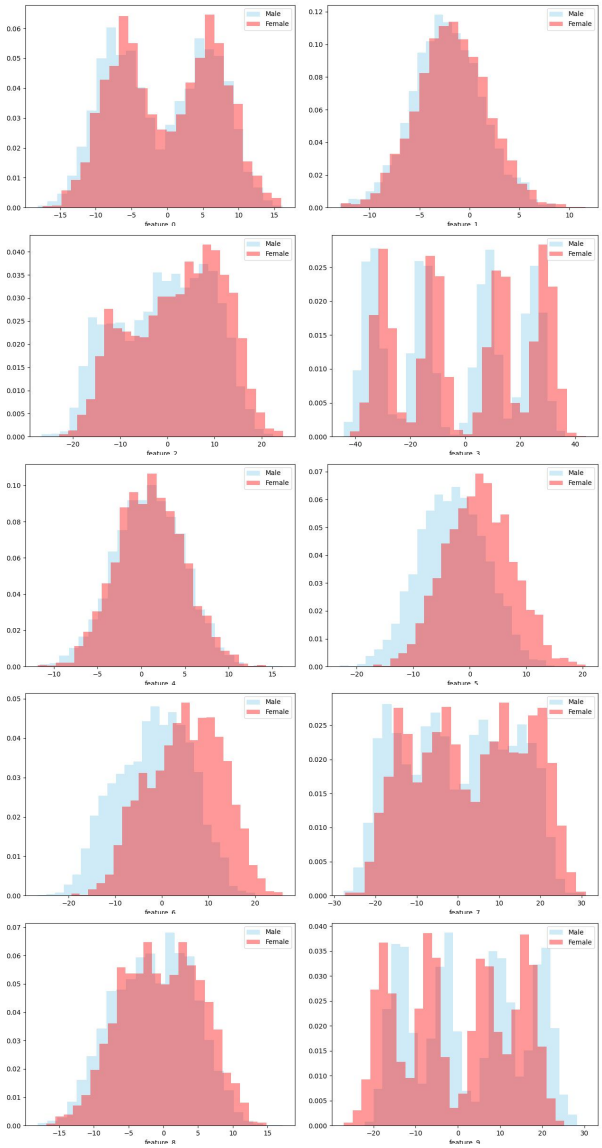
1.2 Exploratory Data Analysis

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.



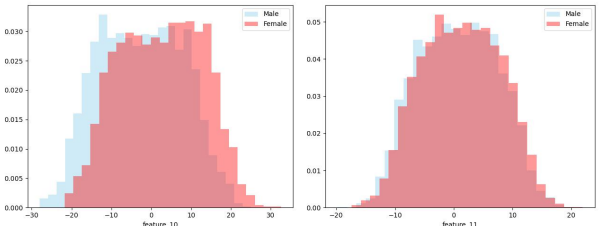


Figure 1: Raw Features

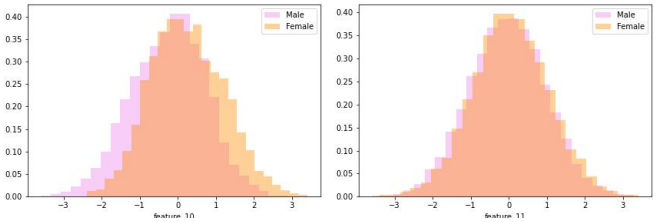
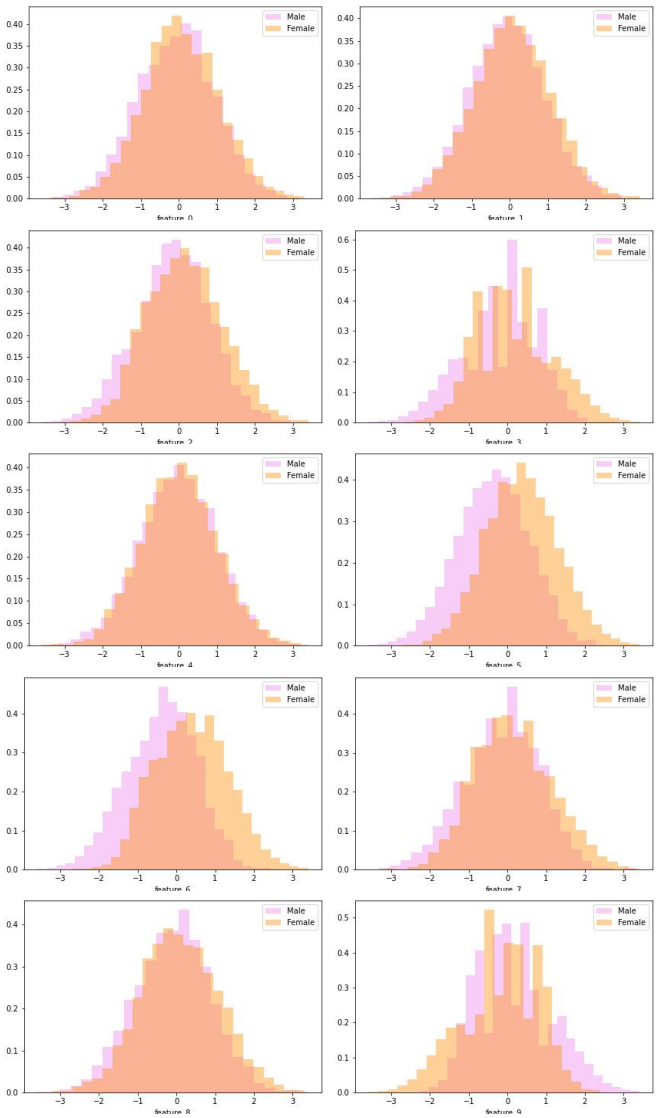


Figure 2: Gaussianized Features

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.



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.

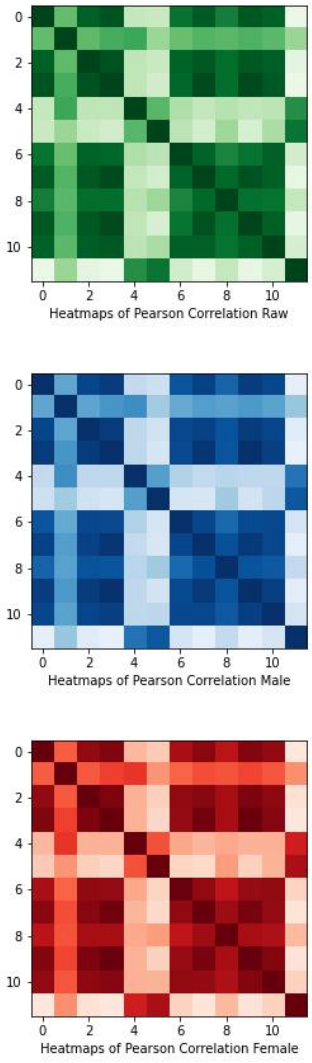


Figure 3: : Heatmap - Pearson Correlation

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.

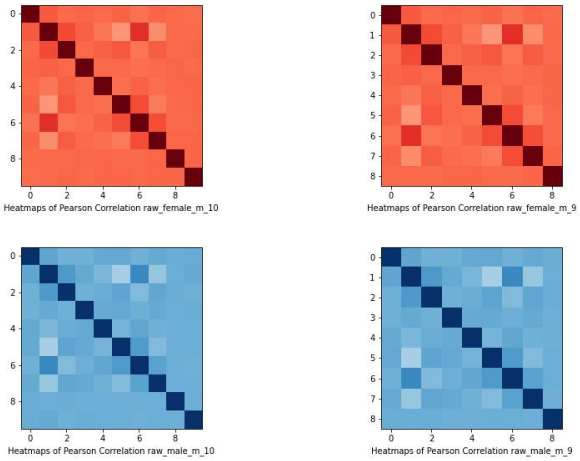


Figure 4: : Heatmap - Pearson Correlation with PCA m=10 and PCA m=9

2 Classification

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

- Multivariate Gaussian Classifier (MVG)
- Logistic Regression
- Support Vector Machine
- Gaussian Mixture Models

K-Fold cross validation is applied to all the models in this paper, with K=5 in order to clarify which model is most promising.

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.

– MVG, Full Covariance - Untied
– MVG, Naive Bayes - Untied
– MVG, Full Covariance - Tied
– MVG, Naive Bayes - Tied

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 $\pi = 0.5$) with π=0.5, π=0.1 and π=0.9:
no PCA, PCA-m=10 and PCA-m=10
5 folds

Table 1: RAW Features, no PCA

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
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

Table 2: Z-Normalized Features, no PCA

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
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

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

PCA gives a good result even with m=9. It was

Table 3: Gaussianized Features, no PCA

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
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

Table 4: RAW Features, PCA m = 10

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
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

Table 5: Z-Normalized Features, PCA m = 10

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
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,

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
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

Table 6: RAW Features, PCA m = 9

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
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

Table 7: Z-Normalized Features, PCA m = 9

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
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

expected since correlation is obvious between features (from heat-maps). 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.

Table 8: Gaussianized Features, PCA m = 9

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
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

2.2 Logistic Regression

Logistic Regression Classifier is a discriminative model. In this report Logistic Regression applied with linear separation rules and Quadratic Logistic Regression will be employed. The implemented objective function is:

$$R(w) = \frac{\lambda}{2}\|w\|^2 + \frac{\pi_T}{n_T}\sum_{i|z_i=1} l(z_i s_i) + \frac{1-\pi_T}{n_F}\sum_{i|z_i=-1} l(z_i s_i)$$

Logistic Regression in this version applies linear separation. However, we can also define non-linear separation by building a specific expanded feature space defined as:

$$\phi(\boldsymbol{x}) = \begin{bmatrix} \text{vec}\left(\boldsymbol{x}\boldsymbol{x}^T\right) \\ \boldsymbol{x} \end{bmatrix}$$

In this version of Logistic Regression model, we are allowed to estimate quadratic separation surfaces in the original space.

It is expected to obtain a good result by Logistic Regression since 'Full Covariance - Tied' performs linear separation rules and consequently it provided a good result. Moreover, since Logistic Regression does not require a particular assumptions on the data distribution, we do not expect 'Z-normalization' and 'Gaussianization' to provide a remarkable improvement on the result.

All the parameters used to obtain results below are :

m: None, 11, 10

gaussianization: "no", "yes"

normalization: "no", "yes"

folds: 5,

pT: [0.5, 0.1, 0.9],

pi: [0.5, 0.1, 0.9],

l: 1e-4

Table 9: LogReg, Raw Features, no PCA

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
LR(λ=10^{-5},πT=0.5)	0.047	0.131	0.126
LR(λ=10^{-5},πT=0.1)	0.047	0.136	0.122
LR(λ=10^{-5},πT=0.9)	0.047	0.130	0.130

PCA m = 11

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
LR(λ=10^{-5},πT=0.5)	0.048	0.133	0.126
LR(λ=10^{-5},πT=0.1)	0.046	0.138	0.120
LR(λ=10^{-5},πT=0.9)	0.048	0.132	0.124

PCA m = 10

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
LR(λ=10^{-5}, πT=0.5)	0.048	0.139	0.122
LR(λ=10^{-5}, πT=0.1)	0.048	0.145	0.121
LR(λ=10^{-5}, πT=0.9)	0.049	0.140	0.124

Table 10: LogReg, Gaussianized Features, no PCA

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
LR(λ=10^{-5},πT=0.5)	0.056	0.163	0.158
LR(λ=10^{-5},πT=0.1	0.055	0.172	0.166
LR(λ=10^{-5},πT=0.9	0.057	0.170	0.160

PCA m = 11

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
LR(λ=10^{-5},πT=0.5)	0.068	0.185	0.188
LR(λ=10^{-5},πT=0.1	0.068	0.191	0.197
LR(λ=10^{-5},πT=0.9	0.068	0.191	0.197

PCA m = 10

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
LR(λ=10^{-5},πT=0.5)	0.068	0.193	0.196
LR(λ=10^{-5},πT=0.1	0.067	0.194	0.202
LR(λ=10^{-5},πT=0.9	0.070	0.192	0.204

Table 11: LogReg, Z-Normalized Features, no PCA

	$\pi = 0.5$	π= 0.1	$\pi = 0.9$
LR(λ=10^{-5},πT=0.5)	0.047	0.131	0.126
LR(λ=10^{-5},πT=0.1)	0.047	0.137	0.122
LR(λ=10^{-5},πT=0.9)	0.048	0.130	0.130

PCA m=11,

	π= 0.5	$\pi = 0.1$	$\pi = 0.9$
LR(λ=10^{-5},πT=0.5)	0.096	0.261	0.224
LR(λ=10^{-5},πT=0.1)	0.098	0.260	0.222
LR(λ=10^{-5},πT=0.9)	0.096	0.266	0.229

PCA m=10

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
LR(λ=10^{-5},πT=0.5)	0.113	0.294	0.260
LR(λ=10^{-5},πT=0.1)	0.113	0.301	0.257
LR(λ=10^{-5},πT=0.9)	0.112	0.299	0.258

2.3 Support Vector Machine

The Support Vector Machine is a discriminative classifier. An SVM is a supervised learning algorithm that is used for classification and regression tasks. It aims to do a hyperplane separation of two classes with the maximum margin. It works by finding the hyperplane in a high-dimensional space that maximally separates the data points into different classes. The model: Linear SVM needs an hyper-parameter tuning for C via cross-validation while employing different values of C for the two classes.

In this paper, 3 different SVMs will consider:
- Linear Support Vector Machine
- Polynomial quadratic kernel degree=2
- Radial Basis Function kernel formulations

2.3.1 Linear SVM

In the case of a linear SVM, the hyperplane is a linear boundary that separates the data points: The linear SVM model is searching for a hyperplane with largest margin and a hyper plane that correctly separates as many instances as possible. The problem is that you will not always able to get both. The C parameter determines how great your desire is for finding a hyperplane that correctly separates as many instances as possible. (low C : large margin, high C : much smaller margin). Therefore C parameter defines a trade-off .

The following graphs show how minDCF varies depending on different values of the hyperparameter C.

All the parameters used to obtain results below are :

m: None (PCA not applied)

gaussianization: "no",

normalization: "yes",

folds: "5 folds",

mode: "Linear",

K: 1.0,

pT: 0.5

pi: [0.5, 0.1, 0.9],

C: [0.0001, 0.001, 0.01, 0.1, 1]

Setting C=1.0 gives the best results.minDCF values with different values of C is reported above with graphs on normalized and gausianized dataset.

Figure 5: DCF Linear SVM with Normalized features

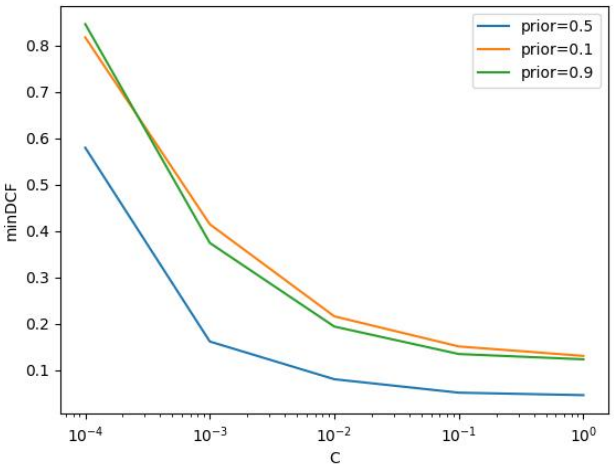


Table 12: Linear SVM, Norm-Features, no PCA, πT=0.5

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
L-SVM, C:0.01, K=1.0	0.080	0.216	0.194
L-SVM, C:0.1, K=1.0	0.051	0.151	0.135
L-SVM, C:1, K=1.0	0.046	0.131	0.123
L-SVM, C:10, K=1.0	0.066	0.190	0.178

Table 13: Linear SVM, Gau-Features, no PCA, πT=0.5

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
L-SVM, C:0.01, K:1.0	0.069	0.190	0.179
L-SVM, C:0.1, K:1.0	0.056	0.165	0.158
L-SVM, C:1, K:1.0	0.055	0.160	0.160
L-SVM, C:10, K:1.0	0.059	0.165	0.160

2.3.2 Quadratic SVM - Polynomial kernel function with degree=2

One of the non-linear SVM model used is the polynomial quadratic kernel:

$$k(x1, x2) = (x1^T x2 + c)^d$$

where d is the degree that is set as 2 (quadratic)

Regularized bias is added to the non - linear SVM as a constant value K to the kernel function.

2.3.3 RBF SVM - Radial Basis kernel SVM

In the context of support vector machines (SVMs), the term "RBF" stands for radial basis function. An RBF

Figure 6: DCF-C graph for Linear SVM with Gausianized features

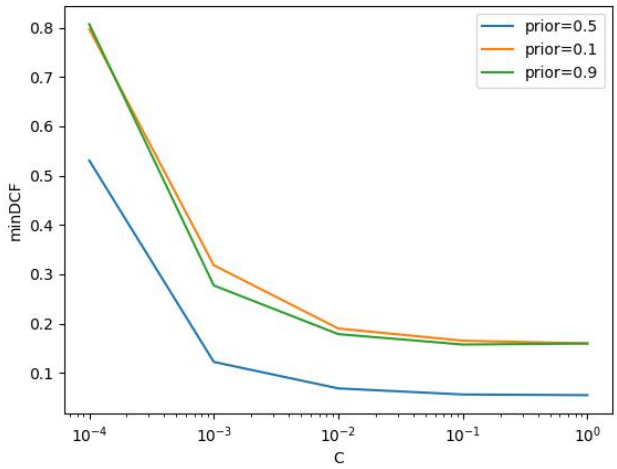
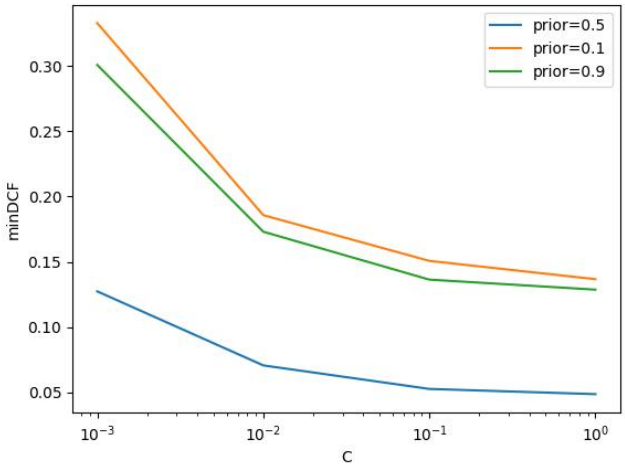


Figure 7: DCF-C graph for Quadratic SVM with Gausianized features



SVM is a type of SVM that uses an RBF kernel to perform non-linear classification.

RBF SVM is capable of finding non-linear boundaries by using an RBF kernel.

The radial basis function (RBF) kernel is a function that measures the similarity between two data points based on their distance from a central point, or "center." The RBF kernel is defined as follows:

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{|\mathbf{x} - \mathbf{y}|^2}{2\sigma^2}\right)$$

where $\mathbf{x}$ and $\mathbf{y}$ are data points, and σ is a free parameter that controls the width of the kernel.

Table 14: Quadratic SVM, Normalized-Features, no PCA, πT=0.5, K=0.1

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Quad SVM, C: 0.01	0.071	0.183	0.171
Quad SVM, C: 0.1	0.053	0.149	0.139
Quad SVM, C :1.0	0.046	0.138	0.128
Quad SVM, C: 10	0.046	0.149	0.135

Quadratic SVM, Normalized-Features, no PCA, πT=0.5, K=1.0

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Quad SVM, C: 0.01	0.071	0.186	0.173
Quad SVM, C: 0.1	0.053	0.151	0.136
Quad SVM, C: 1	0.049	0.137	0.129
Quad SVM, C: 10	0.048	0.141	0.134

In some cases, the RBF kernel is defined with an additional parameter called γ, which is defined as follows:

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\gamma|\mathbf{x} - \mathbf{y}|^2\right)$$

The value of γ determines the shape of the decision boundary and has a significant impact on the performance of the model. A larger value of γ results in a more complex decision boundary, while a smaller value results in a simpler boundary.

The RBF kernel allows the SVM to learn a non-linear decision boundary by mapping the data points to a higher-dimensional space using the kernel function. In this higher-dimensional space, the SVM can find a linear boundary that separates the data points.

Class balancing did not make a huge difference that is why keeping the model without class balancing was considered while results are obtaining.

According to the graph above : gamma-minDCF graph for RBF SVM with Normalized features, best value for γ seems 10^-2 when k=1.0 and c=1.0

Table 15: RBF SVM, Gausianized-Features, no PCA, πT=0.5, gamma : 10^-2

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
RBF SVM, C: 0.1, K=1.0	0.116	0.304	0.285
RBF SVM, C: 1.0, K=1.0	0.082	0.219	0.212
RBF SVM, C: 10, K=1.0	0.146	0.375	0.353

2.3.4 Comparison of well performed Models w.r.t. minDCF:

Comparison of linear models that is discussed in this report is shown below :

Figure 8: gamma-minDCF graph for RBF SVM with Normalized features

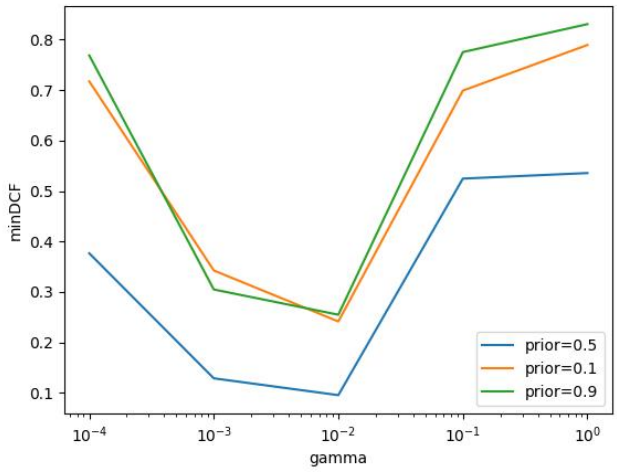


Table 16: Gausianized-Features, no PCA, πT=0.5,

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
MVG (Tied Full-Cov)	0.048	0.125	0.127
Log Reg ($= 10^5$,πT $= 0.5$)	0.047	0.131	0.126
Linear SVM, C=1.0	0.046	0.131	0.123

MVG (Tied Full-Cov)
Log Reg (= 105,T = 0.5)
Linear SVM C=1.0

2.4 Gaussian Mixture Model

GMM is a probabilistic model that assumes observed data is generated from a mixture of several underlying distributions that each is a Gaussian distribution.

To fit a GMM to a dataset, the model's parameters must be estimated from the data. This is typically done by using an iterative algorithm, such as the Expectation-Maximization (EM) algorithm.

The EM algorithm estimates the parameters of the GMM by iteratively updating the estimates of the means, variances, and mixing coefficients of the underlying Gaussian distributions. The GMM is defined by a set of parameters, including:
Mixing coefficients, Means, Covariances:The covariances can be represented using a diagonal matrix or a full covariance matrix.

The GMM is a probabilistic model that assumes that the underlying data is generated from a mixture of a finite number of K Gaussian distributions, where each Gaussian distribution is a component of the model. The com-

ponent parameter in a GMM is the number of Gaussian distributions, or components, that are used to model the data.

In this paper GMM is reported with 2, 4 and 8 components but it could be tested also with higher components. Since validation of the model is computationally very expensive, keeping the number of components to a reasonably small values seems feasible.

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

Table 17: Raw-Features, no PCA, π=0.5,

Components	2	4	8
Full - Covariance	0.043	0.033	0.032
Full - Covariance - Tied	0.043	0.033	0.030
Diag - Covariance	0.400	0.108	0.087
Diag - Covariance - Tied	0.400	0.108	0.082

As a conclusion the best result is obtained with Full - Covariance - Tied GMM with 8 components.

3 Score Calibration

In this part of the report, ROC curve and Bayes Error will be considered.

The Bayes error rate is the lowest possible error rate that can be achieved by a classifier, assuming that the classifier has access to the true underlying probability distribution of the data. In other words, it is the minimum error rate that can be achieved by a classifier when the classifier makes predictions based on the most accurate possible knowledge of the data.

The ROC curve allows you to visually assess the trade-off between the true positive rate and the false positive rate of a classifier, and can be used to compare the performance of different classifiers.

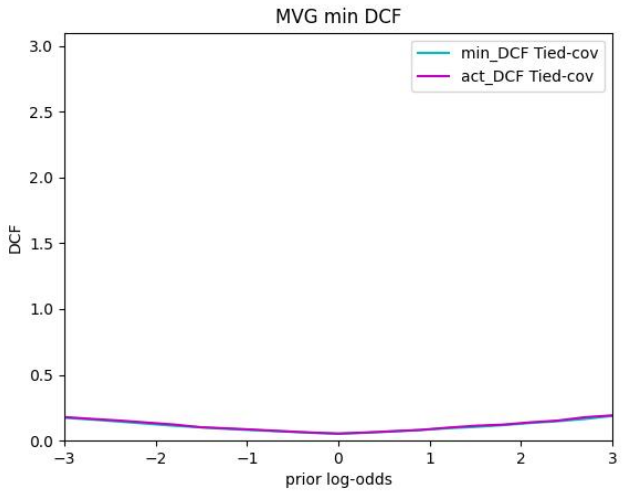
Table 18: Gausianized-Features, no PCA, π=0.5,

Components	2	4	8
Full - Covariance	0.053	0.047	0.048
Full - Covariance - Tied	0.053	0.047	0.050
Diag - Covariance	0.346	0.127	0.106
Diag - Covariance - Tied	0.346	0.127	0.106

3.1 MVG - Score Calibration

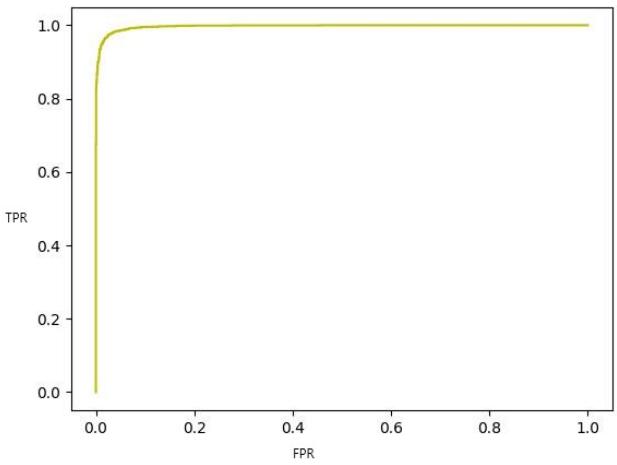
Comparison of minDCF and the actual DCF of the models that have been shown so far in this report.

Figure 9: Bayes Error Plot for MVG - full cov. - tied on Normalized features.



Bayes error plots shows the distance between minDCF and actDCF for MVG full covariance Tied on Normalized features which was one of the best MVG model w.r.t. the results that are obtained.

Figure 10: ROC curve of MVG - full cov. - tied on Normalized features.



A classifier with a higher true positive rate and a lower false positive rate will have a ROC curve that is located

closer to the top left corner of the plot, indicating better performance.
According to the ROC curve our classifier performed well.

3.2 LR - Score Calibration

Bayes error plots show a considerable distance between minDCF and actDCF for Linear Logistic Regression. However according to the ROC curve our classifier performed well.

Figure 11: Bayes Error Plot for Linear LR - on Normalized features.

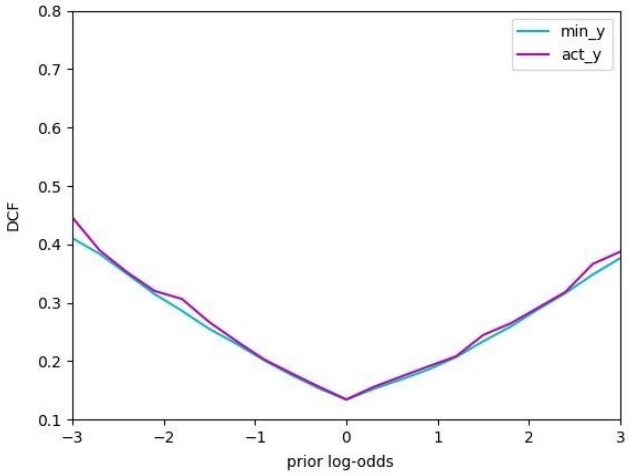
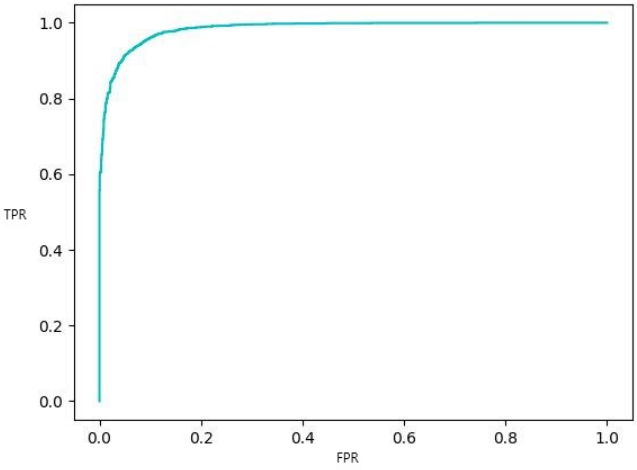


Figure 12: ROC curve Linear LR - on Normalized features.



3.3 SVM - Score Calibration

Bayes error plots shows the distance between minDCF and actDCF for Linear SVM - on Normalized features which was one of the best GMM model w.r.t. the results that are obtained.

Figure 13: Bayes Error of Linear SVM - on Normalized features.

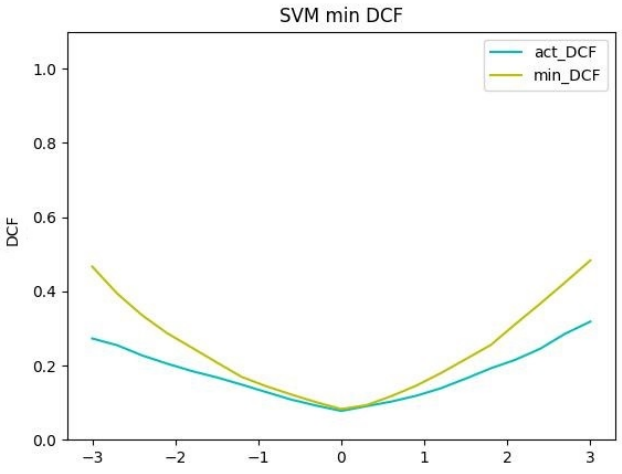
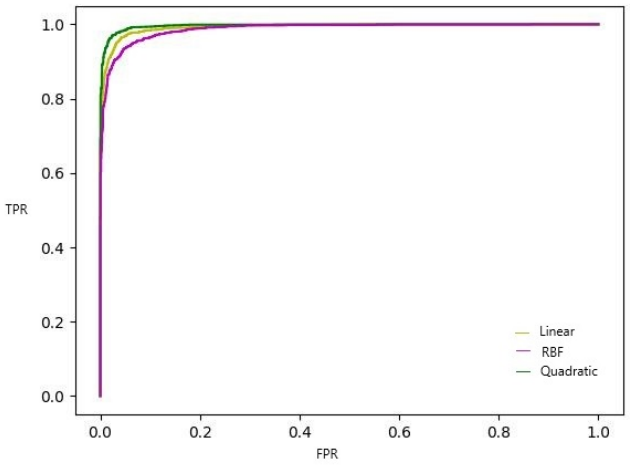


Figure 14: ROC curve of SVM for linear, polynomial and RBF kernel - on Normalized features.

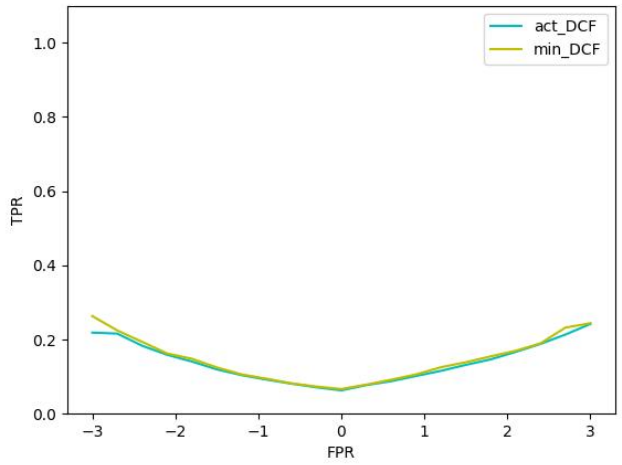


According to the ROC curve our classifier performed well with all the kernels. Linear SVM and quadratic SVM performed similarly well wheres RBF performed worse.

3.4 GMM - Score Calibration

Bayes error plots shows the distance between minDCF and actDCF for GMM with 8 components - on Gausianized features which was one of the best GMM model w.r.t. the results that are obtained.

Figure 15: Bayes Error curve of GMM with 8 components - on Normalized features.



According to the ROC curve our classifier performed well.

4 Experimental results and Conclusion

In this section of the report, the in-so-far best classifiers will be considered with test dataset.
minDCF will be considered to verify the proposed solution can achieve the best accuracy.

Table 19: Norm-Features, no PCA

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
MVG Tied Full Cov	0.050	0.132	0.135
Linear LR	0.052	0.135	0.133
GMM Full-Cov -Tied 8 comp	0.034	0.092	0.086

Test data yielded similar results. However GMM Full Covariance with 8 component seems the best. Therefore it is reasonable to say that Linear models as well as gaussian mixture models proved to fit our dataset.