Gender Detection

Atanasio Giuseppe
Polytechnic of Turin
Artificial Intelligence and Data Analytics
s300733

Carachino Alessio
Polytechnic of Turin
Artificial Intelligence and Data Analytics
s296138

September 2022

Contents

Chapter 1

Introduction

1.1 Abstract

The goal of the application is to build a model that best fits for the Gender Classification task exploiting the most common Machine Learning tools. We will discuss how they perform for the problem we have chosen, explaining pros and cons.

1.2 Problem Overview

The dataset consists of synthetic speaker embeddings that 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 for female, 0 for 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.

1.3 Features Analysis

Here are histograms for each of the 12 features. We can see that the raw features have an approximated gaussian distribution, but in the next page, the features' distributions are plotted after they have been pre-processed (Gaussianize features).

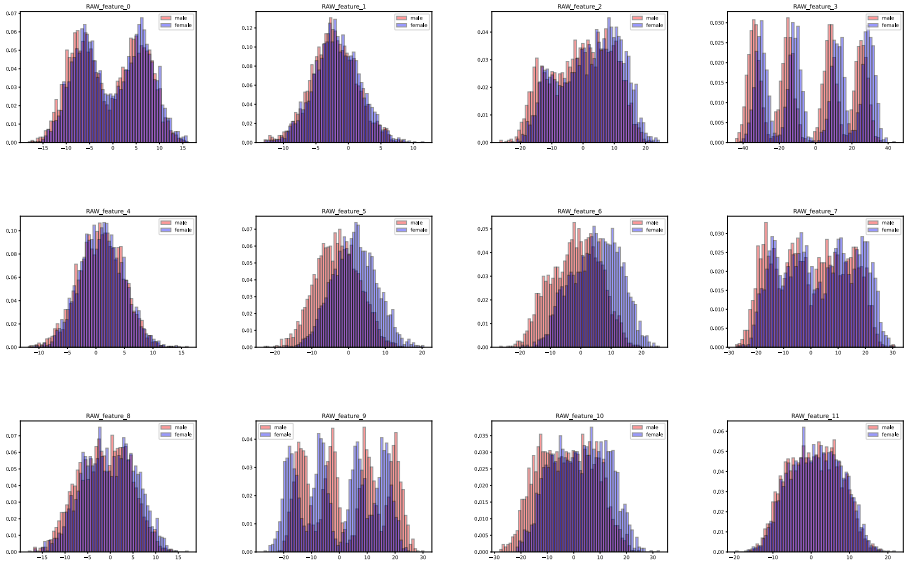
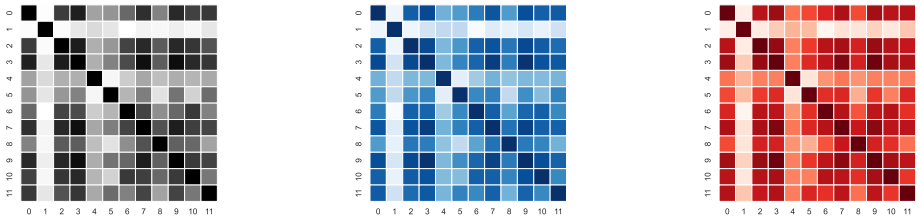


Figure 1.1: RAW Features



(a) All classes (b) Class Male (c) Class Female

Figure 1.2: Heatmap - Pearson Correlation

Feature 3 is highly correlated to the 0, 2, 7 and 9 ones. This suggests we may benefit from using PCA to map data to less correlated features, but we will check later this assumption. Moreover, we can notice that each feature doesn't have outliers, so we don't need to apply them the z-score normalization.

Here are the histograms of the gaussianized features. As we expected, the gaussianization has not brought so much improvement at all. Instead, gaussians in feature 3 and 9 were better in the previous un-preprocessed dataset.

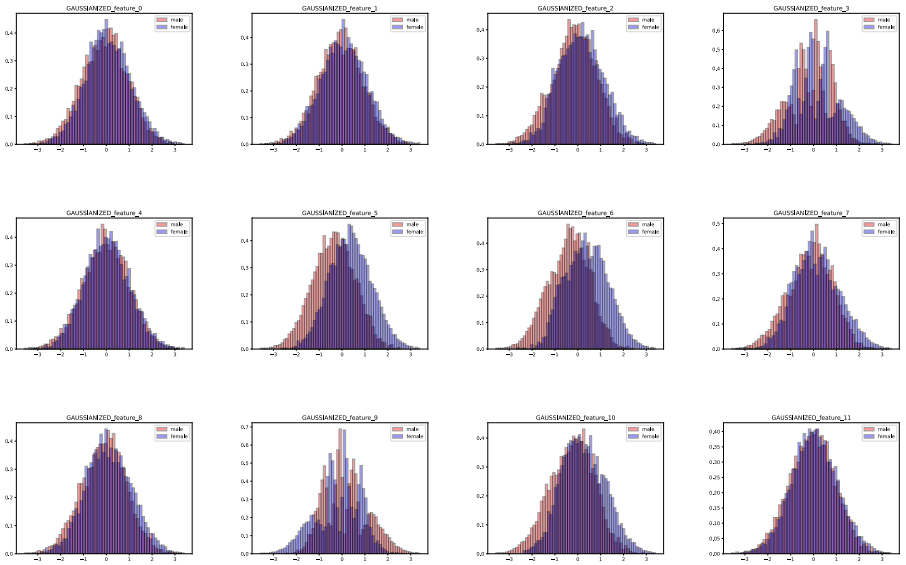
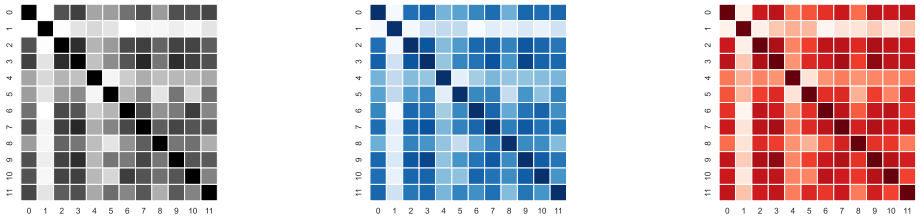


Figure 1.3: Gaussianized Features



(a) All classes (b) Class Male (c) Class Female

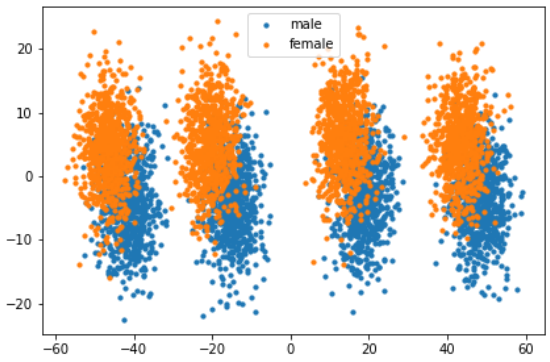
Figure 1.4: Heatmap - Pearson Correlation

Chapter 2

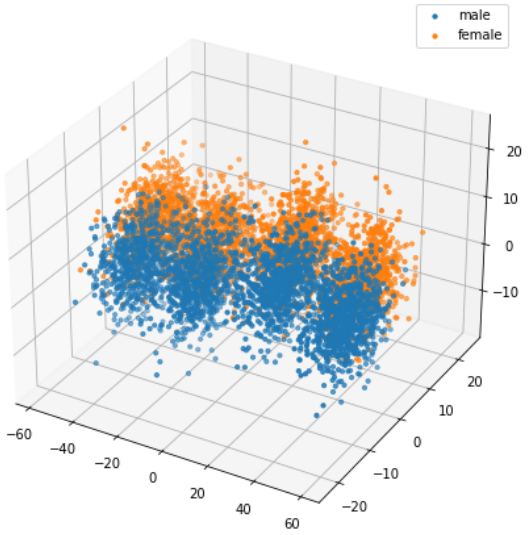
Dimensionality Reduction Techniques

2.1 Principal Component Analysis (PCA)

PCA is the process of computing the principal components and using them to perform a change of basis on the data. This means that it makes it possible to map in another feature space the analysis and even to reduce the number of features of the dataset, reducing for example the training overlay and the risk introduced by the curse of dimensionality. The following images, which are for demonstration purposes only, show the training set after that PCA has been applied.



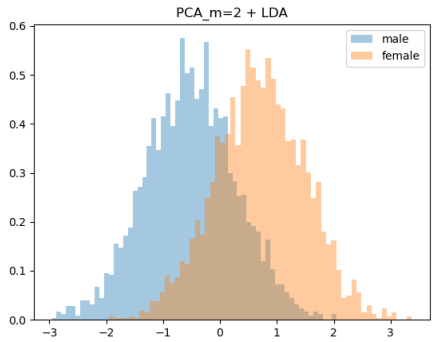
(a) PCA (12 → 2)



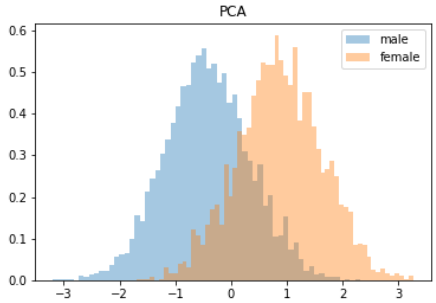
(b) PCA (12 → 3)

2.2 Linear Discriminant Analysis (LDA)

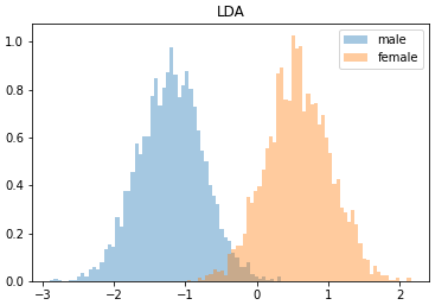
While PCA does not provide any guarantee about obtaining discriminant directions, Linear Discriminant Analysis is defined as a criterion of optimality finds a direction that has a large separation between the classes and small spread inside each class. Also in this case, the following images, which are for demonstration purposes only, show the training set after that LDA has been applied, or PCA followed by LDA, as these two techniques can be used simultaneously (in this strict order).



(a) PCA ($12 \rightarrow 2$) + LDA (histograms show how points are projected along the direction)



(b) PCA ($12 \rightarrow 3$) + LDA (histograms show how points are projected along the direction)



(c) LDA ($12 \rightarrow 1$)

Chapter 3

Classification and Validation

3.1 Introduction

In order to write the report, the collected results have been taken as the output of the following **classifying models**' list:

- 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 considerç
  - a balanced case (our application):

$$(\pi, C_{fn}, C_{fp}) = (0.5, 1, 1)$$

  - two unbalanced cases:

$$(\pi, C_{fn}, C_{fp}) = (0.1, 1, 1)$$
$$(\pi, C_{fn}, C_{fp}) = (0.9, 1, 1)$$

3.2 Multivariate Gaussian Classifiers

We are going to focus the attention on Multivariate Gaussian Classifiers (MVG), in particular for those with the following covariance matrices: Full Covariances, Tied Covariance, Diagonal Covariances. These are generative models with Gaussian distributed data, given the class, as follows:

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

Tied MVG assumes that each class has its own mean μ_c as the other MVG models, but the same covariance matrix for all the classes:

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

The diagonal model assumes that covariance matrices are diagonal matrices.

3.2.1 Expectations

Since histograms have shown that features approximately have a gaussian distribution, it is expected that the Generative Models work well for this dataset. Furthermore, heatmaps have shown that correlation is significantly spreaded between the features. Therefore, it is expected that the models based on the Naïve Bayes assumptions will perform badly.

3.2.2 Results

Here are the results of Gaussian Classifiers in three different applications (ours has $\pi = 0.5$).

	$\pi = 0.5$	$\pi = 0.9$	$\pi = 0.1$	$\pi = 0.5$	$\pi = 0.9$	$\pi = 0.1$	$\pi = 0.5$	$\pi = 0.9$	$\pi = 0.1$
	RAW Features			Gaussianization			Z-Norm		
					no PCA				
FULL-COV	0.048	0.125	0.128	0.062	0.171	0.181	0.048	0.123	0.126
DIAG-COV	0.563	0.856	0.825	0.541	0.824	0.81	0.565	0.848	0.818
TIED FULL-COV	0.047	0.128	0.118	0.06	0.167	0.18	0.048	0.127	0.125
TIED DIAG-COV	0.564	0.85	0.829	0.538	0.816	0.804	0.566	0.845	0.821
					PCA (m=10)				
FULL-COV	0.047	0.124	0.14	0.071	0.204	0.206	0.112	0.263	0.304
DIAG-COV	0.067	0.156	0.162	0.085	0.223	0.228	0.119	0.275	0.294
TIED FULL-COV	0.047	0.125	0.13	0.071	0.206	0.199	0.111	0.259	0.298
TIED DIAG-COV	0.063	0.149	0.153	0.083	0.225	0.227	0.118	0.272	0.293
					PCA (m=9)				
FULL-COV	0.047	0.125	0.139	0.091	0.238	0.242	0.158	0.376	0.403
DIAG-COV	0.065	0.153	0.164	0.095	0.258	0.26	0.161	0.377	0.417
TIED FULL-COV	0.047	0.123	0.131	0.09	0.233	0.236	0.155	0.367	0.398
TIED DIAG-COV	0.062	0.146	0.153	0.096	0.261	0.26	0.161	0.376	0.415

3.2.3 Considerations

As it was expected, PCA brings good results even with 9 features and improves models based on Naïve assumption, but they aren't remarkable with respect to the full and tied RAW features. So, for the discriminative approaches, it is not expected that PCA works well; however, we reported the results anyway.

Tied Models works well, and this suggests that models which exploits linear separation rules are expected to perform effectively. This will be confirmed later.

Lastly, Z-score Normalized features does not bring remarkable results. We can notice that the ones with diagonal covariance matrices perform worse than full covariance ones, and this is due to the highly correlation between features, as we have assumed above with the expectations.

3.3 Logistic Regression

3.3.1 Objective Function

Since we want to consider different applications (with different priors), we have implemented the prior-weighted version of the Logistic Regression. In other words, the implemented objective function is:

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

This version of Logistic Regression applies linear separation rules, but it is possible to define non-linear separation rules by building a certain expanded feature space defined as:

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

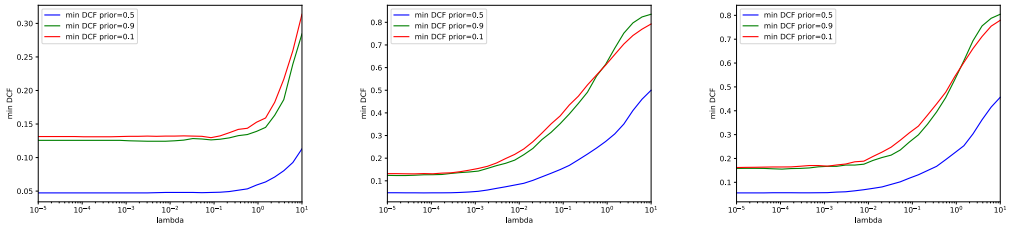
We can thus train the LR model defined above, but this time using the feature vectors $\phi(x)$ rather than x. It allows computing linear separation rules for $\phi(x)$, and this corresponds to estimate quadratic separation surfaces in the original space.

3.3.2 Expectations

Since 'TIED FULL-COV' introduces linear separation rules and has provided good results, it is expected that Logistic Regression will perform well. Furthermore, we do not expect that with 'Gaussianization' or 'Z-score Normalization' we will obtain significantly better results as Logistic Regression does not require specific assumptions on the data distribution.
Lastly, for what concerns the Quadratic version of the Logistic Regression, it is not expected for regularization to have a significant impact on the minDCF computation due to the higher complexity of the model.

3.3.3 Results

The plots show how minDCF is affected by different values of λ. They are exploited to calibrate λ, which is the regularization term.



(a) DCF - RAW LogReg (b) DCF - Z-norm. LogReg (c) DCF - Gauss. LogReg

The tuning of the hyper-parameter λ shows that regularization is required and brings benefits, with best results obtained for small values of it. Moreover, for $\lambda = 1$, the results significantly starts to get worse, especially for Z-score Normalization and Gaussianization of the features. The scale normalization effects of Z-score and Gaussianization do not seem much relevant, and it is needed a smaller regularization term than Logistic Regression with RAW features. PCA provides almost the same results as the ones without it.

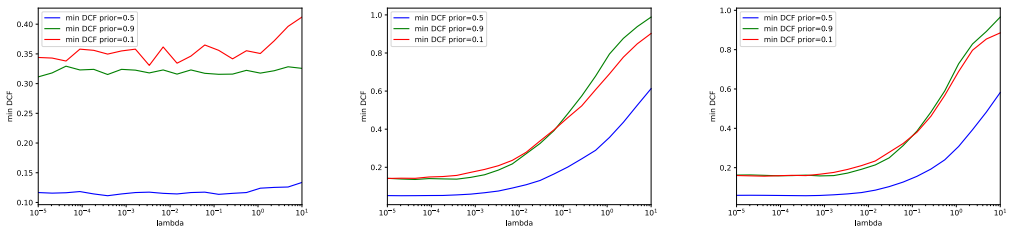
We now turn the attention on the tabular values for different π_T: the outcomes are very similar.

	RAW			Gauss			Znorm		
	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
no PCA									
LogReg($\lambda = 10^{-5}, \pi_T = 0.5$)	0.048	0.131	0.125	0.056	0.163	0.159	0.047	0.132	0.123
LogReg($\lambda = 10^{-5}, \pi_T = 0.1$)	0.046	0.138	0.121	0.054	0.173	0.166	0.047	0.139	0.123
LogReg($\lambda = 10^{-5}, \pi_T = 0.9$)	0.047	0.132	0.129	0.057	0.171	0.161	0.047	0.13	0.129
PCA (m=10)									
LogReg($\lambda = 10^{-5}, \pi_T = 0.5$)	0.049	0.139	0.122	0.068	0.194	0.2	0.112	0.297	0.26
LogReg($\lambda = 10^{-5}, \pi_T = 0.1$)	0.049	0.145	0.121	0.067	0.195	0.204	0.114	0.302	0.256
LogReg($\lambda = 10^{-5}, \pi_T = 0.9$)	0.049	0.14	0.125	0.07	0.194	0.203	0.112	0.3	0.261

We can conclude that MVG with Tied Covariance Matrix performs similarly. It is reasonably to affirm that Logistic Regression retrieve the same outcomes of the MVG model with linear classification rules. Moreover, different values of π_T does not bring improvements for the other two applications.

Since MVG Tied model also performs quite similar to the MVG ones that follows a quadratic separation rule, we repeat the analysis for Quadratic Logistic Regression.

3.4 Quadratic Logistic Regression



(a) DCF - RAW QLogReg (b) DCF - Znorm. QLogReg (c) DCF - Gauss. QLogReg

Regularization is not so helpful for RAW features, but it is for pre-processed features with Z-Score Normalization and Gaussianization as in the previous model.

Again, let's consider training with different values of π_T:

	RAW			Gauss			Znorm		
	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
no PCA									
QLogReg($\lambda = 10^{-5}, \pi_T = 0.5$)	0.117	0.344	0.311	0.059	0.16	0.162	0.051	0.141	0.142
QLogReg($\lambda = 10^{-5}, \pi_T = 0.1$)	0.129	0.357	0.352	0.059	0.158	0.174	0.055	0.15	0.144
QLogReg($\lambda = 10^{-5}, \pi_T = 0.9$)	0.153	0.472	0.342	0.061	0.177	0.172	0.053	0.148	0.142
PCA (m=10)									
QLogReg($\lambda = 10^{-5}, \pi_T = 0.5$)	0.049	0.139	0.122	0.068	0.194	0.2	0.112	0.297	0.26
QLogReg($\lambda = 10^{-5}, \pi_T = 0.1$)	0.049	0.145	0.121	0.067	0.195	0.204	0.114	0.302	0.256
QLogReg($\lambda = 10^{-5}, \pi_T = 0.9$)	0.049	0.14	0.125	0.07	0.194	0.203	0.112	0.3	0.261

This model gets worse results than the previous ones only for RAW features. Pre-processed dataset with PCA $m = 10$, has the same outcomes as the Logistic Regression. Using different values of π_T does not improve the classification performance for the specific application, and the results for imbalanced applications are pretty similar to each other, even though $\pi_T = 0.5$ performs better than the others.

3.4.1 Conclusion

We can conclude that classes are better separated with linear decision rules. The study for this application continues with the training of SVMs and to Gaussian-Mixture Models, and we start with the first one for which we will expect good results.

To confirm our previous assumptions, we continue with the pre-processing techniques analyzed in the previous models (Z-score Normalization and Gaussianization).

3.5 Support Vector Machine

3.5.1 About SVM...

The implemented SVM models are:

- **Linear SVM**, obtained by solving the primal problem, expressed as the minimization of:

$$\widehat{J}(\widehat{\boldsymbol{w}}) = \frac{1}{2}\|\widehat{\boldsymbol{w}}\|^2 + C\sum_{i=1}^{n}\max\left(0, 1 - z_i\left(\widehat{\boldsymbol{w}}^T\widehat{\boldsymbol{x}}_i\right)\right)$$

  where

$$\widehat{\boldsymbol{x}}_i = \begin{bmatrix} \boldsymbol{x}_i \\ K \end{bmatrix}, \quad \widehat{\boldsymbol{w}} = \begin{bmatrix} \boldsymbol{w} \\ b \end{bmatrix}$$

  and K is a regularization term which mitigates the effects of the bias term due to the regularization of the norm of $\widehat{\boldsymbol{w}}$. In fact, it is equal to $\|\widehat{\boldsymbol{w}}\|^2 = \|\boldsymbol{w}\|^2 + b^2$.

- **Quadratic SVM**, obtained by solving the dual problem, expressed as the maximization of:

$$\alpha^T\mathbf{1} - \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{H}\boldsymbol{\alpha}$$

  subject to

$$0 \leq \alpha_i \leq C, i = 1, \ldots, n, \quad \sum_{i=1}^{n}\alpha_i z_i = 0$$

  Directly solving the formulated expression would increase the scores computation complexity. This is the reason why a kernel function is employed. Specifically, it computes the dot product in the expanded space $\boldsymbol{\Phi}\left(\boldsymbol{x}_i\right)^T\boldsymbol{\Phi}\left(\boldsymbol{x}_j\right)$. Formally, the kernel function is then defined as $k\left(\boldsymbol{x}_1, \boldsymbol{x}_2\right) = \boldsymbol{\Phi}\left(\boldsymbol{x}_1\right)^T\boldsymbol{\Phi}\left(\boldsymbol{x}_2\right)$. Actually, in order to add a regularized bias in the non-linear SVM, it is sufficient to add a constant value $\xi = K^2$ to the kernel function:

$$\widehat{k}\left(x_1, x_2\right) = k\left(x_1, x_2\right) + \xi$$

  It can be computed in different ways. For the polynomial kernel of degree d, it is

$$k\left(x_1, x_2\right) = \left(x_1^T x_2 + 1\right)^d$$

- **SVM** with **R**adial **B**asis kernel **F**unction, obtained by solving the same dual problem as above but, in this case, the kernel function is defined as

$$k\left(\boldsymbol{x}_1, \boldsymbol{x}_2\right) = e^{-\gamma\|x_1 - x_2\|^2}$$

  where γ is an hyper-parameter.

3.5.2 Expectations

Up to now, models which exploit linear separation rules have performed very well. So it is expected that Linear SVM will provide good results as well.

3.5.3 Results

The plots show how minDCF is affected by different values of the hyper-parameter of C. They have been generated, as in the previous models, employing the k-fold with k=5.

Linear SVM — RAW Features

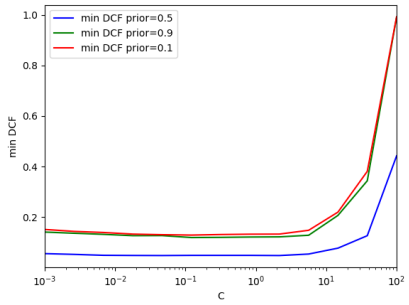


Figure 3.3: DCF for Linear SVM with RAW features (K=1.0)

Setting C=1.0 seems a reasonable choice. However, we reported minDCF values for a set of different values of C anyways, just to assess about the consistency of plot and values.

		$\pi = 0.5$	
	K = 0.1	K = 1.0	K = 10.0
C=0.01	0.06	0.049	0.048
C=0.1	0.057	0.048	0.048
C=1.0	0.051	0.048	0.047
C=10.0	0.093	0.096	0.065
		$\pi = 0.1$	
C=0.01	0.163	0.135	0.132
C=0.1	0.161	0.13	0.13
C=1.0	0.14	0.133	0.138
C=10.0	0.252	0.232	0.165
		$\pi = 0.9$	
C=0.01	0.151	0.13	0.124
C=0.1	0.146	0.12	0.118
C=1.0	0.129	0.119	0.127
C=10.0	0.291	0.291	0.157

Linear SVM — Gaussianized Features

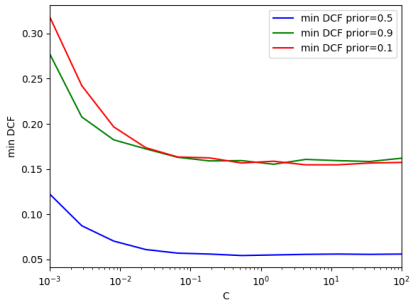


Figure 3.4: DCF for Linear SVM with Gaussianized features (K=1.0)

	$\pi = 0.5$		
	K = 0.1	K = 1.0	K = 10.0
C=0.01	0.068	0.069	0.069
C=0.1	0.057	0.056	0.056
C=1.0	0.054	0.054	0.054
C=10.0	0.056	0.056	0.055
	$\pi = 0.1$		
C=0.01	0.19	0.19	0.189
C=0.1	0.167	0.165	0.165
C=1.0	0.156	0.156	0.156
C=10.0	0.154	0.153	0.158
	$\pi = 0.9$		
C=0.01	0.179	0.178	0.179
C=0.1	0.156	0.156	0.156
C=1.0	0.16	0.16	0.16
C=10.0	0.16	0.16	0.16

Linear SVM — Z-Score Normalized Features

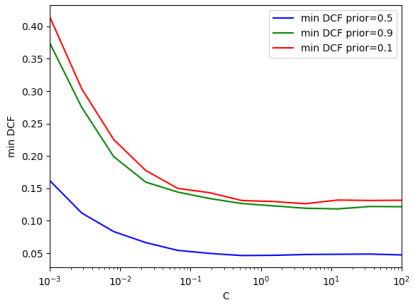


Figure 3.5: DCF for Linear SVM with Z-score normalized features (K=1.0)

	$\pi = 0.5$		
	K = 0.1	K = 1.0	K = 10.0
C=0.01	0.08	0.081	0.081
C=0.1	0.05	0.051	0.051
C=1.0	0.047	0.046	0.046
C=10.0	0.048	0.048	0.048
	$\pi = 0.1$		
C=0.01	0.216	0.215	0.217
C=0.1	0.149	0.15	0.15
C=1.0	0.13	0.131	0.131
C=10.0	0.129	0.129	0.128
	$\pi = 0.9$		
C=0.01	0.192	0.194	0.194
C=0.1	0.139	0.139	0.139
C=1.0	0.123	0.124	0.124
C=10.0	0.121	0.121	0.118

As far as pre-processing techniques, it seems that they do not have a significant impact on Linear SVM results.

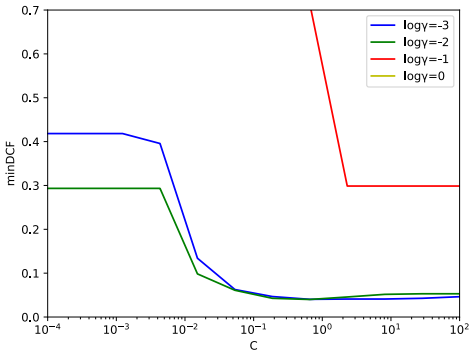
Quadratic SVM — Polynomial kernel function with degree=2

C = 1.0, D=2	K = 1.0	K = 10.0
	Raw features	
$\pi = 0.5$	0.721	0.722
$\pi = 0.1$	0.999	1.0
$\pi = 0.9$	1.002	0.992
	Gaussianized features	
$\pi = 0.5$	0.061	0.062
$\pi = 0.1$	0.167	0.168
$\pi = 0.9$	0.171	0.171
	Z-Score normalized features	
$\pi = 0.5$	0.053	0.052
$\pi = 0.1$	0.149	0.149
$\pi = 0.9$	0.146	0.146

Unlike the previous case, pre-processing techniques have a significant impact on Quadratic SVM. In this case, the best results have been obtained with z-score normalized features and $\xi = K^2 = 10^2$. The results provided by different values of C are very similar and not reported.

SVM with Radial Basis kernel Function — RAW Features

The following plot is different from the ones generated in the other SVM cases. In particular, this plot allows to tune two hyper-parameters at the same time (C and γ).



In this case, C and γ influence the results. Fortunately, the results are stabilized after a certain C value for both $\log \gamma = -3$ and $\log \gamma = -2$. On the other hand, they get worse for greater values of γ. Let's continue the analysis for $C = 1.0$ and $\log \gamma = -3$.

$\gamma = 0.001, C = 1.0$	K = 0.1	K = 1.0	K = 10.0
$\pi = 0.5$	0.039	0.039	0.04
$\pi = 0.1$	0.123	0.123	0.12
$\pi = 0.9$	0.12	0.12	0.12

For what concerns K, it does not have a significant impact on the results at all.

3.5.4 Considerations

We can compare linear models in terms of minDCF:

	$\pi = 0.5$	$\pi = 0.9$	$\pi = 0.1$
MVG (Tied Full-Cov)	0.047	0.128	0.118
(W)Log Reg ($\lambda = 10^{-5}$, $\pi_T = 0.5$)	0.048	0.131	0.125
Linear SVM, C=1.0	0.048	0.119	0.133

Linear SVM performs similarly to other linear approaches, as expected. Since Linear models perform similarly to non-linear models, we decided to report the results for the two SVM non-linear formulations.

3.6 Gaussian Mixture Model

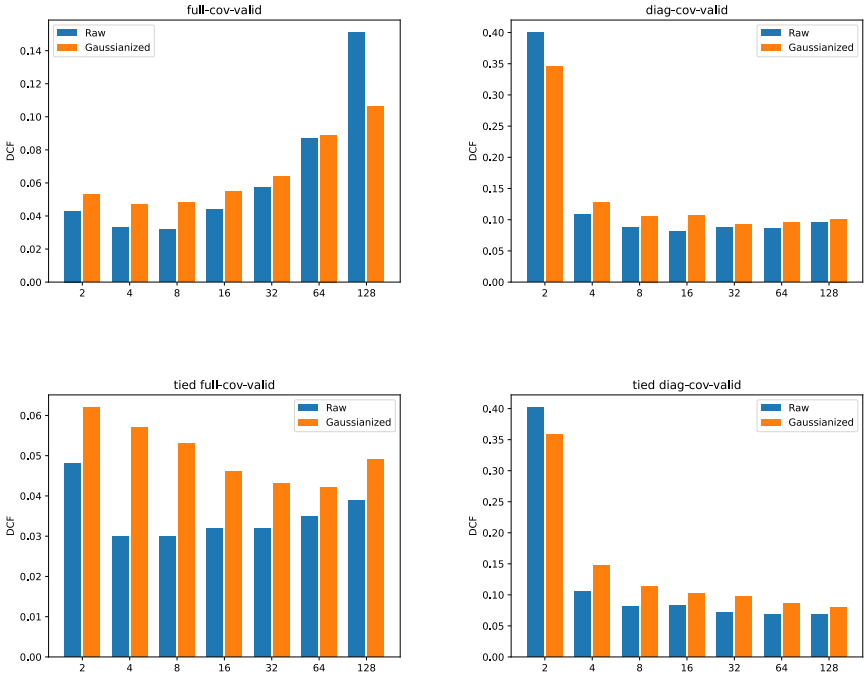
The last model we consider is Gaussian Mixture Model (GMM), in which the hyperparameter that has to be tuned is the number of Gaussian Components, both for RAW and Gaussianized Features.

3.6.1 Expectations

GMMs can approximate generic distributions, so we expect to obtain better results than with the Gaussian model. We are going to train GMM up to 128 components for full and diagonal covariance, and tied/non tied.

3.6.2 Results

Components	2	4	8	16	32	64	128
				RAW			
Full-Cov	0.043	0.033	0.032	0.044	0.057	0.087	0.151
Diag-Cov	0.4	0.108	0.087	0.081	0.088	0.086	0.095
Tied Full-Cov	0.048	0.03	0.03	0.032	0.035	0.035	0.039
Tied Diag-Cov	0.402	0.105	0.082	0.083	0.072	0.068	0.069
				Gaussianized			
Full-Cov	0.053	0.047	0.048	0.055	0.064	0.089	0.106
Diag-Cov	0.346	0.127	0.106	0.107	0.093	0.096	0.101
Tied Full-Cov	0.062	0.057	0.053	0.046	0.043	0.042	0.049
Tied Diag-Cov	0.358	0.148	0.113	0.103	0.098	0.086	0.08

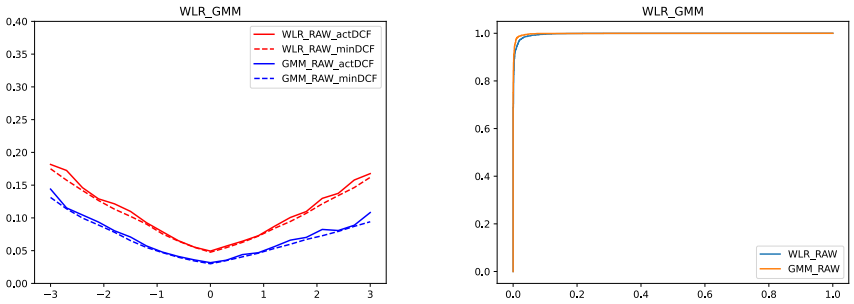


3.6.3 Considerations

We can check that the best result that it could be obtained is for full-covariance tied GMM, especially the 4-D one. Taking a zoom for the 4-D models, we can analyse bayes error plots: the best is again the full-covariance tied.

3.7 The wrap up

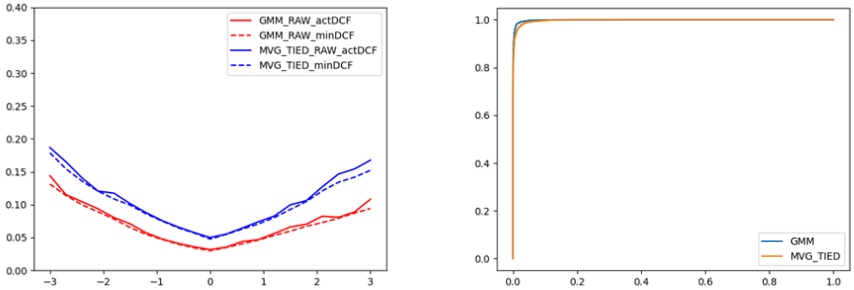
We still should compare the models. It's time to exploit ROC curves and Bayes Error Plots. Firstly, here is the comparison (based on RAW features) between the GMM with 4 components and Tied Full Covariance Matrix and Logistic Regression:



(a) Bayes Error Plot (b) ROC curve

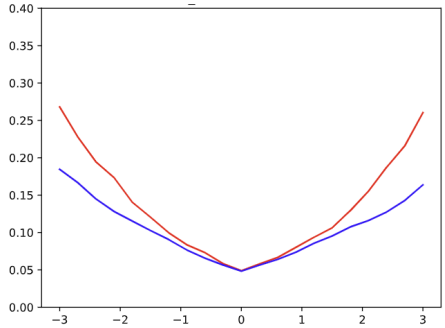
It is clear that, in terms of both minDCF and actDCF, GMM outperforms Logistic Regression. However, calibration is not required as the actDCF curve is already quite close to the minDCF curve.

Now, let's turn the attention on the comparison (RAW features in this case too) between MVG with Tied Full Covariance and the same GMM as above.



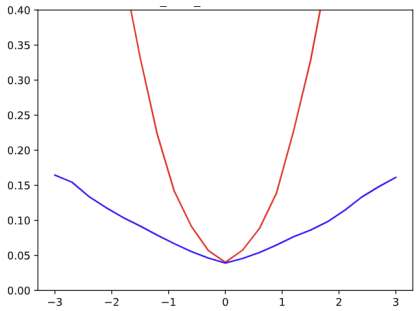
(a) Bayes Error Plot (b) ROC curve

It turns out that the exact same considerations of the previous comparison are still valid.
For what concerns SVM, the following plot shows the Bayes Error Plots for the case of the linear SVM:



(a) Bayes Error Plot(model + hyperparametes...)

The above results are good but not better than the previously compared models. Furthermore, the actDCF curve indicates that scores are mis-calibrated. It was an expected fact, since SVM scores have no probabilistic interpretation. This is even more remarkable in the following case where Radial Basis kernel Function has been used:



(a) Bayes Error Plot(Uncalibrated SVM-RBF, C=1.0, $\log\gamma$=-3, RAW features)

In this case, scores are highly uncalibrated. It means that they can be improved by applying a transformation function f to previously computed scores, in order to obtain calibrated scores $s_{cal} = f(s)$. To estimate f, discriminative score models, such as Logistic Regression, can be used by passing the scores to the model. They would act as if they were a feature of the model. Following this path, $f(s)$ can be interpret as the log-likelihood ratio:

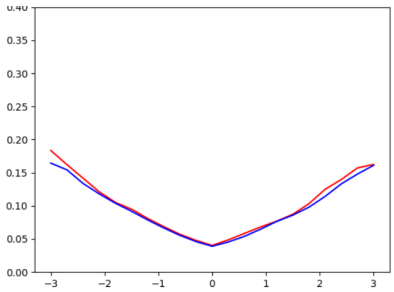
$$f(s) = \log \frac{f_{S|C}\left(s \mid H_T\right)}{f_{S|C}\left(s \mid H_F\right)} = \alpha s + \beta$$

while the class posterior probability for a given prior $\tilde{\pi}$ would be equal to:

$$\log \frac{P\left(C = H_T \mid s\right)}{P\left(C = H_F \mid s\right)} = \alpha s + \beta + \log \frac{\tilde{\pi}}{1-\tilde{\pi}}$$

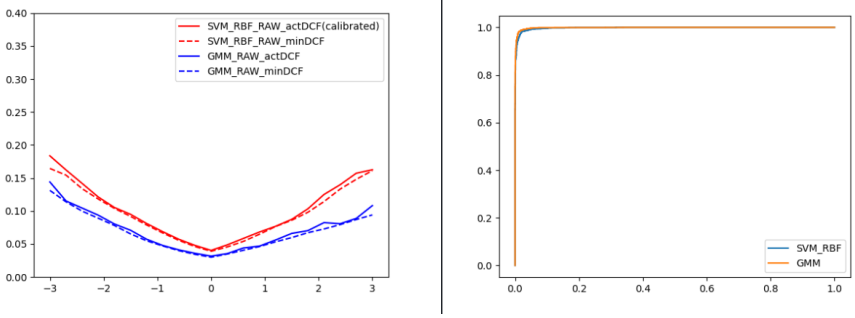
where $\tilde{\pi}$ is a model paramer which represents the application we are optimizing the scores for. Finally, to retrieve the calibrates scores, $f(s)$ can thus be computed as:

$$f(s) = \alpha s + \beta = \alpha s + \beta' - \log \frac{\tilde{\pi}}{1-\tilde{\pi}}$$



(a) Bayes Error Plot(SVMRBF with calibrated scores, $\tilde{\pi}$=0.5, C=1.0, $\log\gamma$=-3, RAW features)

What follows is a direct comparison between the already analyzed models of SVMRBF and GMM:



(a) Bayes Error Plot

(b) ROC curve

Final Pick before using the test set

We expect that the **4D-GMM Tied Full Covariance Model** outperforms the others.

Chapter 4

Experimental Results

4.1 Introduction

We now turn the attention to the evaluation part. It is useful to choose the **minimum DCF** to evaluate because it is an optimistic estimation for the actual DCF in the hypothetical case in which it could be chosen the optimal threshold for the evaluation set. Since validation has been performed with 5-fold cross-validation, the evaluation part will be executed using the whole dataset (all training data for training, all test data for test).

4.2 Multivariate Gaussian Classifiers

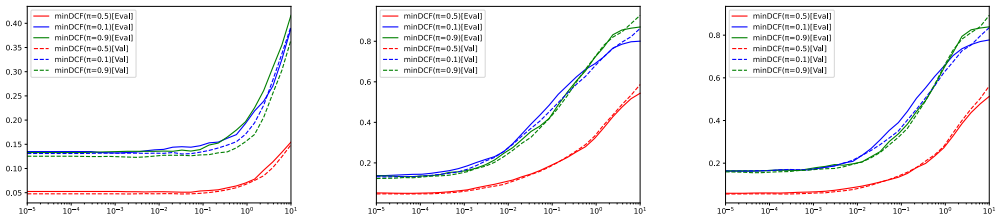
Here are reported the evaluation results for MVG:

	$\pi = 0.5$	$\pi = 0.9$	$\pi = 0.1$	$\pi = 0.5$	$\pi = 0.9$	$\pi = 0.1$
	RAW features			Gaussianization		
	no PCA					
Full-Cov	0.053	0.138	0.134	0.073	0.182	0.202
Diag-Cov	0.57	0.882	0.81	0.547	0.846	0.792
Tied Full-Cov	0.05	0.135	0.132	0.069	0.177	0.184
Tied Diag-Cov	0.57	0.88	0.808	0.545	0.846	0.793
	PCA (m=10)					
Full-Cov	0.052	0.139	0.136	0.08	0.206	0.203
Diag-Cov	0.068	0.18	0.198	0.088	0.229	0.243
Tied Full-Cov	0.054	0.134	0.138	0.078	0.205	0.199
Tied Diag-Cov	0.068	0.181	0.196	0.085	0.225	0.234
	PCA (m=9)					
Full-Cov	0.054	0.14	0.144	0.098	0.23	0.255
Diag-Cov	0.068	0.18	0.204	0.102	0.248	0.29
Tied Full-Cov	0.054	0.136	0.142	0.098	0.232	0.258
Tied Diag-Cov	0.068	0.18	0.2	0.104	0.258	0.286

The results are consistent with those obtained in the validation part. We can confirm that the MVG model with tied covariance is the best option to choice, and this also confirms that linear separation rules are the best picks. PCA is not outperforming with respect to the previous, even if it improves the MVG models with diagonal covariance. Gaussianization worsen the data.

4.3 Logistic Regression

We now turn our attention to the Logistic Regression models with estimated $\lambda = 10^{-5}$:



(a) DCF - RAW LogReg (b) DCF - Z-norm. LogReg (c) DCF - Gauss. LogReg

For what concerns the DCF curves, they confirms our expectations since validation and evaluation ones have the same trend, and confirms that our choice of $\lambda = 10^{-5}$ has given quite remarkable results.

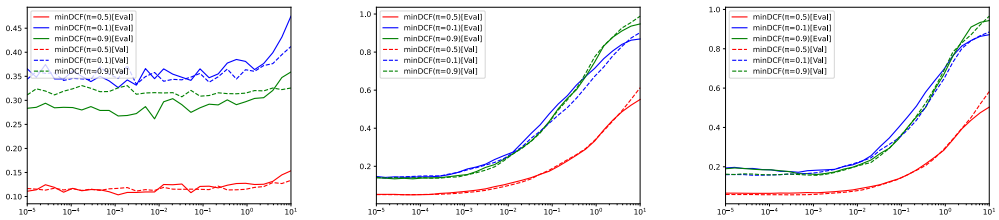
We now move on the analysis for different values of π_T:

	RAW			Gauss			Znorm		
	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
no PCA									
LogReg($\lambda = 10^{-5}, \pi_T = 0.5$)	0.053	0.135	0.133	0.059	0.162	0.165	0.052	0.138	0.136
LogReg($\lambda = 10^{-5}, \pi_T = 0.1$)	0.052	0.138	0.136	0.062	0.166	0.166	0.051	0.14	0.136
LogReg($\lambda = 10^{-5}, \pi_T = 0.9$)	0.053	0.142	0.132	0.062	0.168	0.165	0.052	0.144	0.131
PCA (m=10)									
LogReg($\lambda = 10^{-5}, \pi_T = 0.5$)	0.054	0.146	0.136	0.079	0.2	0.212	0.13	0.313	0.295
LogReg($\lambda = 10^{-5}, \pi_T = 0.1$)	0.052	0.144	0.135	0.079	0.194	0.219	0.132	0.308	0.295
LogReg($\lambda = 10^{-5}, \pi_T = 0.9$)	0.053	0.15	0.14	0.08	0.204	0.215	0.13	0.316	0.292

Results are consistent with our expectations. Linear models performs similar to the MVG linear classifiers. The outcomes are slightly worse but very similar than the expectations, and changing target prior is not helpful at all since they are similar to each other, with a little improvement for $\pi_T = 0.1$ but not significant.

4.4 Quadratic Logistic Regression

We repeat the analysis for Quadratic Logistic Regression:



(a) DCF - RAW LogReg (b) DCF - Z-norm. LogReg (c) DCF - Gauss. LogReg

Gaussianization and Z-score Normalization features analysis retrieves the same trend both on the evaluation and validation set. We can conclude that our choice of $\lambda = 10^{-5}$ was effective also in this case.

We now turn the attention to the minDCF for different values of λ on the evaluation set.

	RAW			Gauss			Znorm		
	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
no PCA									
QLogReg($\lambda = 10^{-5}, \pi_T = 0.5$)	0.111	0.366	0.284	0.066	0.194	0.19	0.052	0.146	0.14
QLogReg($\lambda = 10^{-5}, \pi_T = 0.1$)	0.16	0.396	0.353	0.066	0.177	0.193	0.054	0.148	0.154
QLogReg($\lambda = 10^{-5}, \pi_T = 0.9$)	0.173	0.488	0.33	0.068	0.207	0.194	0.054	0.154	0.133
PCA (m=10)									
LogReg($\lambda = 10^{-5}, \pi_T = 0.5$)	0.054	0.146	0.136	0.079	0.2	0.212	0.13	0.313	0.295
LogReg($\lambda = 10^{-5}, \pi_T = 0.1$)	0.052	0.144	0.135	0.079	0.194	0.219	0.132	0.308	0.295
LogReg($\lambda = 10^{-5}, \pi_T = 0.9$)	0.053	0.15	0.14	0.08	0.204	0.215	0.13	0.316	0.292

As we expected, the outcomes for the evaluation set are consistent with those on the validation one. Also in this case RAW features retrieve quite worse results than Logistic Regression. Z-score Normalization and Gaussianization retrieves similar results with respect to the previous model, and PCA with $m = 10$ are equal to the Logistic Regression ones for all type of features as in the validation set analysis. Even in this case, $\pi_T = 0.5$ performs better than the others.

For RAW features the curves are similar for our application and $\pi_T = 0.1$, but for $\pi_T = 0.9$ the results on the evaluation set are quite better than on the validation set.

4.5 Support Vector Machine

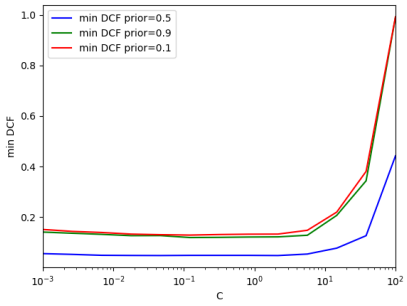
Briefly, for each implemented version of SVM, the selected values for the model parameters, during the previous phase, are:

- Linear SVM: RAW features, C=1.0, K=1.0
- Quadratic SVM: Z-score normalized features, (d=2), C=1.0, K=10.0
- SVM RBF: RAW features, C=1.0, K=1.0, $log\gamma = -3$

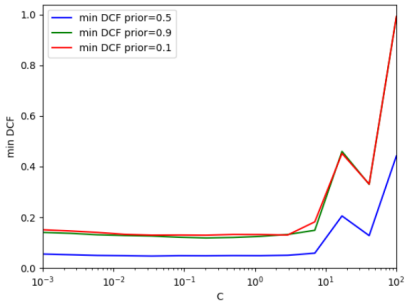
	$\pi = 0.5$	$\pi = 0.9$	$\pi = 0.1$
Linear SVM	0.051	0.129	0.138
Quadratic SVM	0.053	0.144	0.139
SVM RBF	0.044	0.112	0.133

SVM RBF clearly performs better than its counterparts in terms of minDCF.
Let's turn the attention to assessing whether the selected hyper-parameters values have been right or not, we can compare side-by-side their results on both the validation set(left) and evaluation set(right):

- Linear SVM:

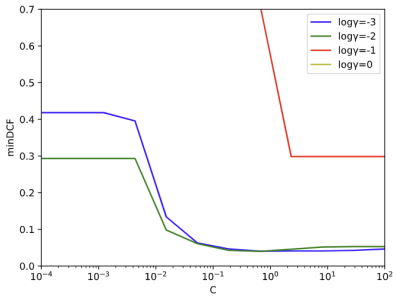


(a) Validation

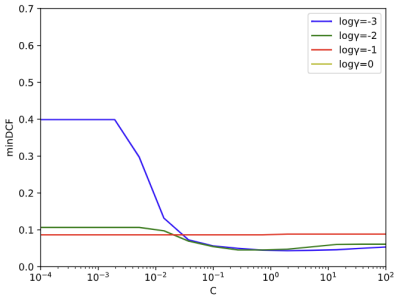


(b) Evaluation

- SVM RBF:



(a) Validation

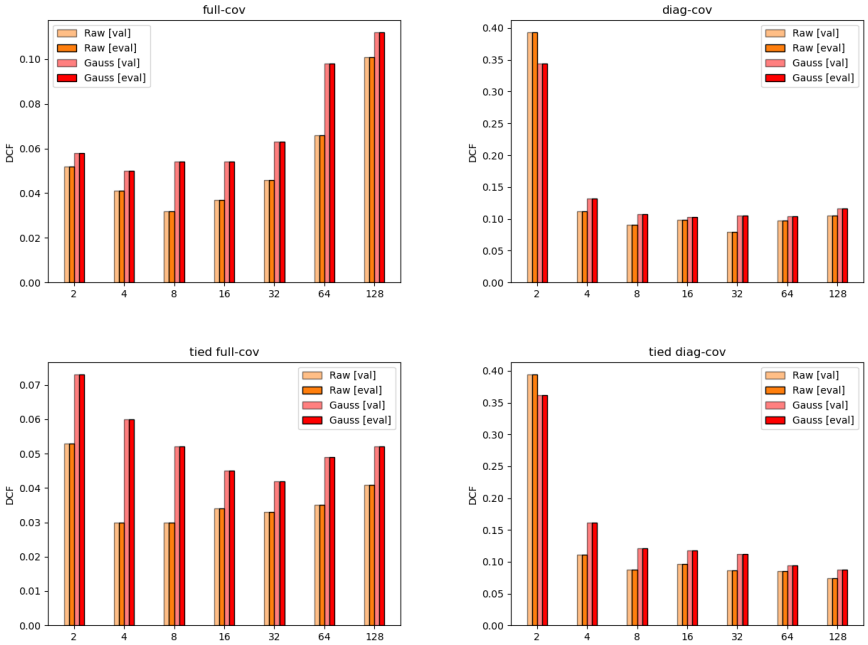


(b) Evaluation

4.6 Gaussian Mixture Model

We have chosen 4 components results to analyze with other applications.

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
	RAW		
Full-Cov	0.041	0.114	0.119
Diag-cov	0.112	0.258	0.29
Tied Full-Cov	0.03	0.08	0.092
Tied Diag-Cov	0.111	0.256	0.295
	Gaussianization		
Full-Cov	0.05	0.149	0.147
Diag-cov	0.132	0.328	0.357
Tied Full-Cov	0.06	0.16	0.176
Tied Diag-Cov	0.161	0.411	0.399

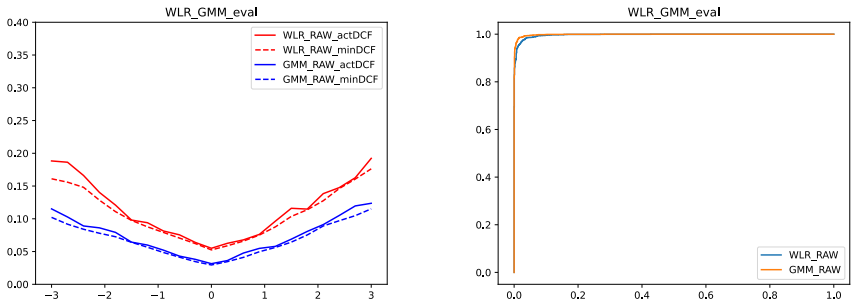


The outcomes are consistent with those in the validation part. We can confirm that linear models outperform, especially for the 4-D GMM tied full covariance case. It is remarkable that full-covariance models over the 8-D tends to have more problems of overfitting than the diagonal ones.

4.7 Choosing best two models

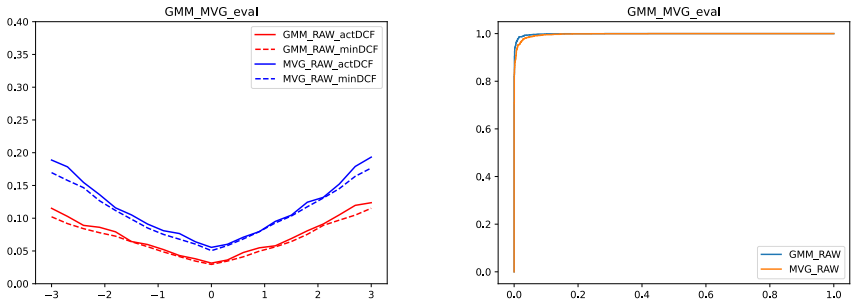
As in the validation phase, we still should compare the models in terms of minDCF and actDCF using Bayes Error Plots (on the left) and ROC curves (on the right).

Firstly, here is the comparison (based on RAW features) between the GMM with 4 components and Tied Full Covariance Matrix and Logistic Regression with $\lambda = 10^{-5}$:



Even in the evaluation phase, GMM outperforms Logistic Regression.

We now move on the comparison between MVG with Tied Full Covariance and the same GMM as above:



Also in this case, as in the validation phase, GMM performs better than MVG with Tied Covariance Matrix.

For what concerns SVM RBF:

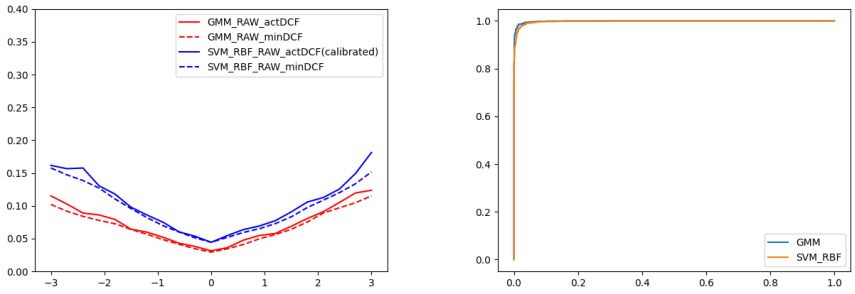


Figure 4.8: SVM RBF with calibrated scores, $\tilde{\pi}$=0.5, C=1.0, $\log\gamma$=-3, RAW features)

Again, GMM outperforms all the previous model as expected, confirming the results obtained on the validation set.

	minDCF	actDCF	minDCF	actDCF	minDCF	actDCF
	$\pi = 0.5$		$\pi = 0.1$		$\pi = 0.9$	
GMM 4 comp. Tied Full	0.03	0.032	0.08	0.088	0.092	0.096

Chapter 5

Conclusion

In conclusion, our approach, consisting on the choice of the GMM - 4 components with Tied Full Covariance Matrix, has resulted effective.

We are able to achieve a DCF cost of ≈ 0.03 for our main application ($\pi = 0.5$). The model is good also for applications with imbalanced prior $\pi = 0.1$ (DCF cost of ≈ 0.08) and for the one with prior 0.9 (DCF cost of ≈ 0.09). The choices we made on our training / validation sets proved effective also for the evaluation data.