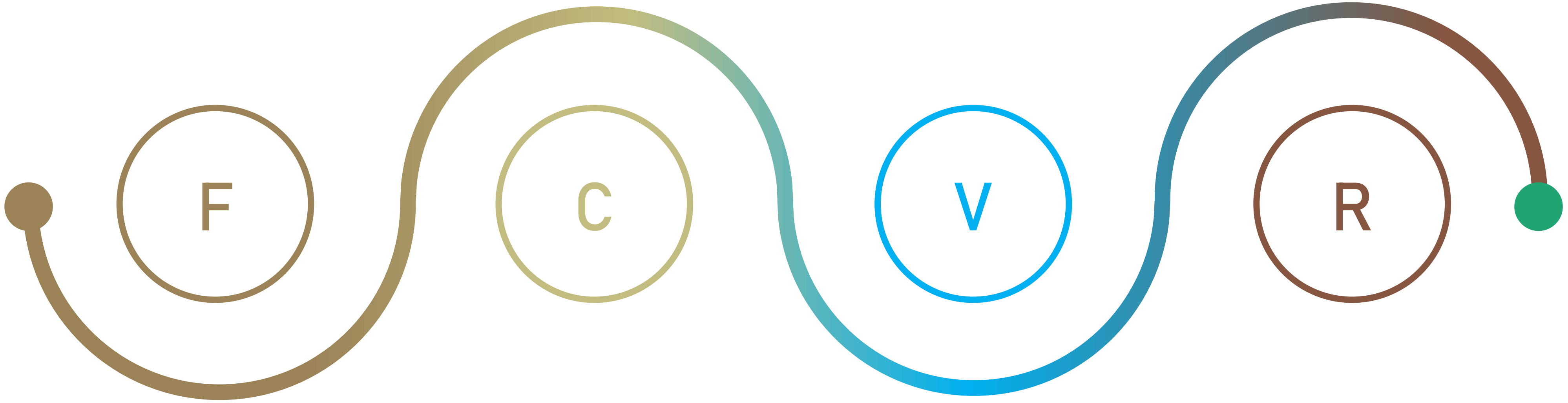
GENDER DETENCTION

Politecnico di Torino

1859

Davide Aiello, Giulia Mannaioli

GENDER DETENCTION



Features analysis Building classifiers Experimental validation Results analysis

Gender detenction dataset

This project consists in gender identification from high-level features.

The dataset consists of synthetic speaker embeddings that represent the acoustic characteristics of a spoken utterance (the dataset consists of synthetic samples that behave similarly to real speaker embeddings).

A speaker embedding is a small-dimensional, fixed sized representation of an utterance.

Speakers belong to four different age groups. The age information, however, is not available.

The training set consists of 3000 samples per class, whereas the test set contains 2000 samples per class.

Classes are balanced.

Gender detenction features

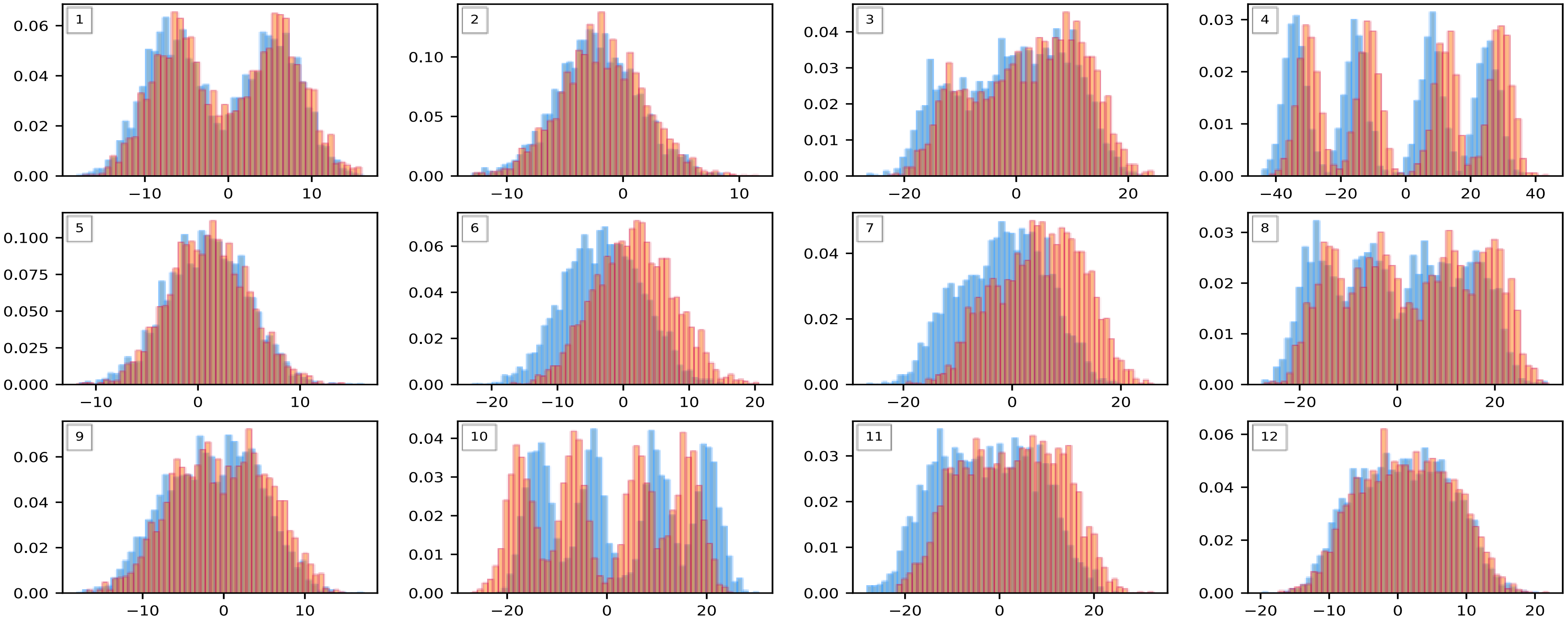
Each sample (each row) corresponds to a different speaker, and contains 12 features followed by the gender label (1 for female, 0 for male).

The features do not have any particular interpretation.

Features are continuous values that represent a point in the mdimensional embedding space.

The embeddings have already been computed.

Histogram of the features



Histogram of the the Gender detection training set features. Features are sorted by their order, from left to right, top to bottom. Red histograms refer to female class, blue histograms to male class.

Gaussianification

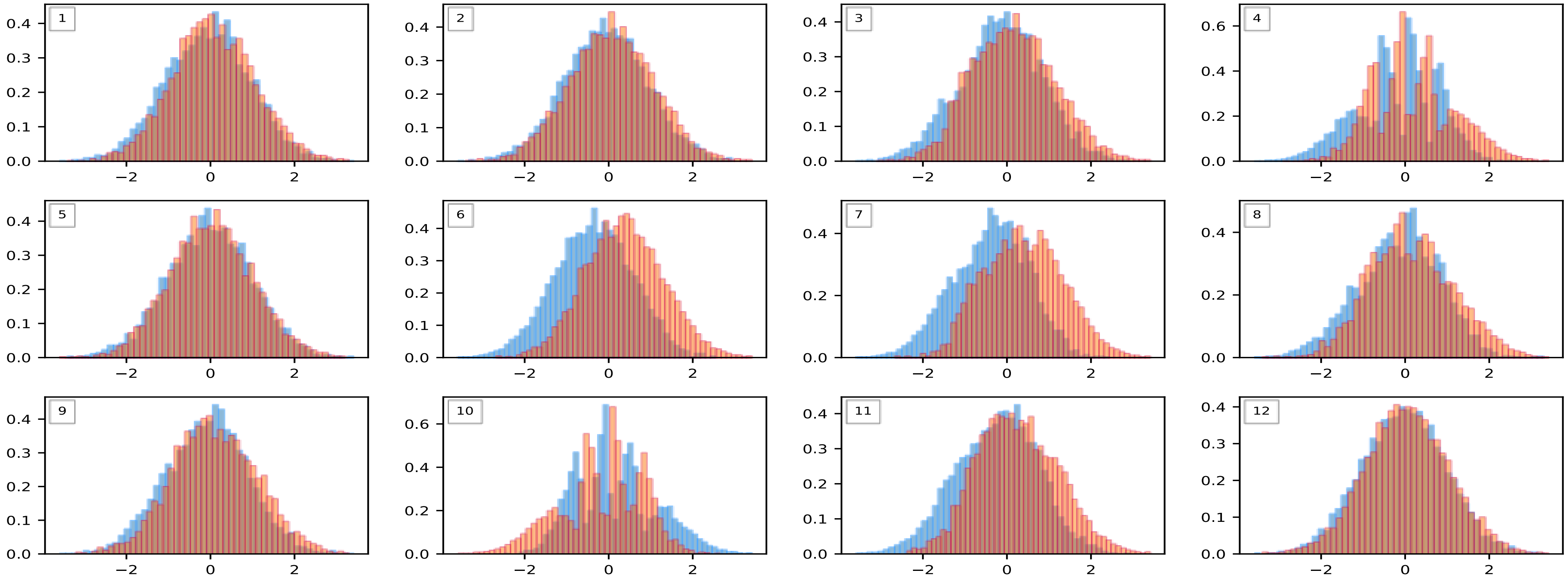
A preliminary analysis of the training data shows that there are not significant outliers but in many cases the raw features do not have gaussian distribution (in particular in 1, 4, 8, 10); this behaviour is related to the age split.

We can also notice that there is a consistent overlapping between the two classes.

The most discriminant features are 6, 7 and 10.

However we will pre-process data by “Gaussianizing” the features.

Gaussianized features histogram



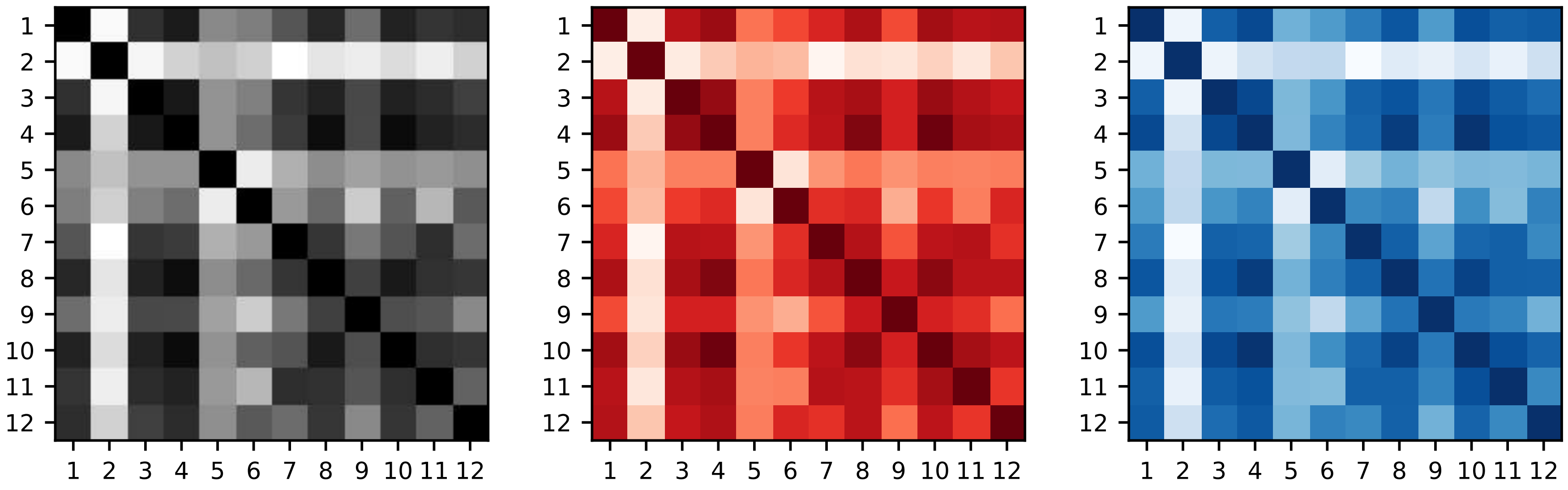
Histogram of the Gender detection training set features after gaussianization.

Features are sorted by their order, from left to right, top to bottom. Red histograms refer to female class, blue histograms to male class.

Correlation matrix

Gray: whole dataset. Red: samples of female. Blue: samples of male.

The three are based on row data.



The correlation matrices revealed a strong correlation between features (in particular 1-4, 3-4, 4-8, 4-10). This suggests we may benefit from using PCA to map data up to 8 uncorrelated features to reduce the number of parameters to estimate.

Furthermore, the within-class covariance matrices are not diagonal. This means that we do not expect to obtain good results with diagonal models.

Anyway, we will consider all the variants of the multivariate gaussian model.

Since 7 features have a similar gaussian distribution, we may have good results with the gaussian classifiers (except for diagonal variants).

As far as the Tied variant model concerns, analyzing the data, we figure out that the two class covariance matrices are similar. In fact, if we compute the difference between one of the two with the within class covariance matrix (which is the parameter used in tied model) of the whole dataset is almost 0. Thus we expect to have good results with this variant of the model.

Choose K-fold number

We chose to employ K-Fold cross-validation to have more data available for training and validation and to obtain more robust results. Data has been shuffled before splitting, so that the data of different folds are homogeneous.

Leave-One-Out

	no Gaussianization (k=6000)		
	$\tilde{\pi}$ = 0.1	$\tilde{\pi}$ = 0.5	$\tilde{\pi}$ = 0.9
RAW DATA (no PCA)			
Full-Cov	0.126	0.048	0.124
Diag-Cov	0.820	0.565	0.856
Tied Full-Cov	0.142	0.047	0.149
Tied Diag-Cov	0.999	0.574	0.997
PCA (m=11)			
Full-Cov	0.129	0.048	0.127
Diag-Cov	0.177	0.065	0.159
Tied Full-Cov	0.149	0.048	0.152
Tied Diag-Cov	0.189	0.065	0.194
PCA (m=10)			
Full-Cov	0.137	0.048	0.120
Diag-Cov	0.173	0.068	0.159
Tied Full-Cov	0.145	0.049	0.156
Tied Diag-Cov	0.198	0.066	0.197

We tried Leave-One-Out and even if it gave us good results it required about 6 hours. Since we have to estimate the hyperparameters for the other models, we decided to use K=5, which seamed to be a good trade-off between time and robustness. Thus, we will not take into account these data.

PCA (m=9)			
Full-Cov	0.135	0.047	0.122
Diag-Cov	0.171	0.068	0.161
Tied Full-Cov	0.151	0.048	0.152
Tied Diag-Cov	0.203	0.065	0.203
PCA (m=8)			
Full-Cov	0.136	0.047	0.120
Diag-Cov	0.169	0.067	0.159
Tied Full-Cov	0.149	0.049	0.151
Tied Diag-Cov	0.204	0.066	0.203

We will consider different values of m for PCA (up to 7) and with/out gaussianizzation.

We also considered different applications (0.1, 0.5, 0.9).

We expect to have worst results with m=7 for what we considered before.

MVG Classifier – min DCF on the validation set

K=5						
	no Gaussianization			Gaussianization		
	$\tilde{\pi}$ = 0.1	$\tilde{\pi}$ = 0.5	$\tilde{\pi}$ = 0.9	$\tilde{\pi}$ = 0.1	$\tilde{\pi}$ = 0.5	$\tilde{\pi}$ = 0.9
RAW DATA (no PCA)						
Full-Cov	0.126	0.048	0.123	0.181	0.062	0.171
Diag-Cov	0.818	0.565	0.848	0.810	0.541	0.824
Tied Full-Cov	0.136	0.047	0.134	0.199	0.060	0.182
Tied Diag-Cov	0.957	0.574	0.964	0.955	0.542	0.954
PCA (m=11)						
Full-Cov	0.132	0.048	0.127	0.212	0.073	0.208
Diag-Cov	0.177	0.066	0.160	0.230	0.086	0.230
Tied Full-Cov	0.210	0.071	0.192	0.220	0.072	0.228
Tied Diag-Cov	0.181	0.064	0.180	0.246	0.083	0.259
PCA (m=10)						
Full-Cov	0.140	0.047	0.120	0.206	0.071	0.204
Diag-Cov	0.173	0.067	0.161	0.228	0.085	0.223
Tied Full-Cov	0.205	0.069	0.205	0.211	0.070	0.231
Tied Diag-Cov	0.180	0.067	0.184	0.243	0.082	0.258

MVG Classifier – min DCF on the validation set

K=5						
	no Gaussianization			Gaussianization		
	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$
PCA (m=9)						
Full-Cov	0.137	0.046	0.121	0.242	0.091	0.238
Diag-Cov	0.171	0.067	0.159	0.260	0.095	0.258
Tied Full-Cov	0.207	0.069	0.206	0.258	0.091	0.272
Tied Diag-Cov	0.184	0.066	0.181	0.277	0.096	0.300
PCA (m=8)						
Full-Cov	0.136	0.046	0.118	0.440	0.166	0.418
Diag-Cov	0.165	0.067	0.157	0.453	0.174	0.422
Tied Full-Cov	0.209	0.069	0.204	0.474	0.166	0.479
Tied Diag-Cov	0.189	0.067	0.180	0.489	0.174	0.498
PCA (m=7)						
Full-Cov	0.278	0.110	0.270	0.432	0.167	0.420
Diag-Cov	0.320	0.120	0.299	0.444	0.177	0.425
Tied Full-Cov	0.359	0.127	0.373	0.476	0.167	0.480
Tied Diag-Cov	0.355	0.122	0.344	0.480	0.173	0.505

The Full-Cov model performs in general better.

As expected, the diagonal models do not perform well with row data but applying PCA, which reduces correlation among the features, the model provides better results. PCA diagonalizes the global covariance matrix, in fact we can see that diagonal methods perform considerably better than with raw data.

PCA does not improve a lot our estimate (however, consistently with our previous observations, PCA with $m > 7$ does not degrade performance, actually $m=8$ and $m=9$ give the best result). Whereas further reduction decreases accuracy we will not consider anymore $m < 8$.

Among the m > 8 the results do not have large variations. We suppose it is because the feature 4 is correlated to four of the other features and probably it is the one preserved among these by PCA because it has the highest variance.

We decide to leave behind Tied model because it is good just on raw.

Gaussianization does not improve performance. Due to this, we decided to not carry on the gaussianized features.

However, the imbalanced tasks are slightly worse than the balanced one .

Overall, the best candidate is currently the MVG model with Full Covariance matrices with m = 8 and with the raw data .

Now, we will focus on discriminative approaches.

Given the limited effectiveness of gaussianizing for generative models we only consider the not pre-processed data.

In this case, we compare the results with raw data and with PCA(m = 8).

Logistic Regression

Let’s start with regularized Logistic Regression.

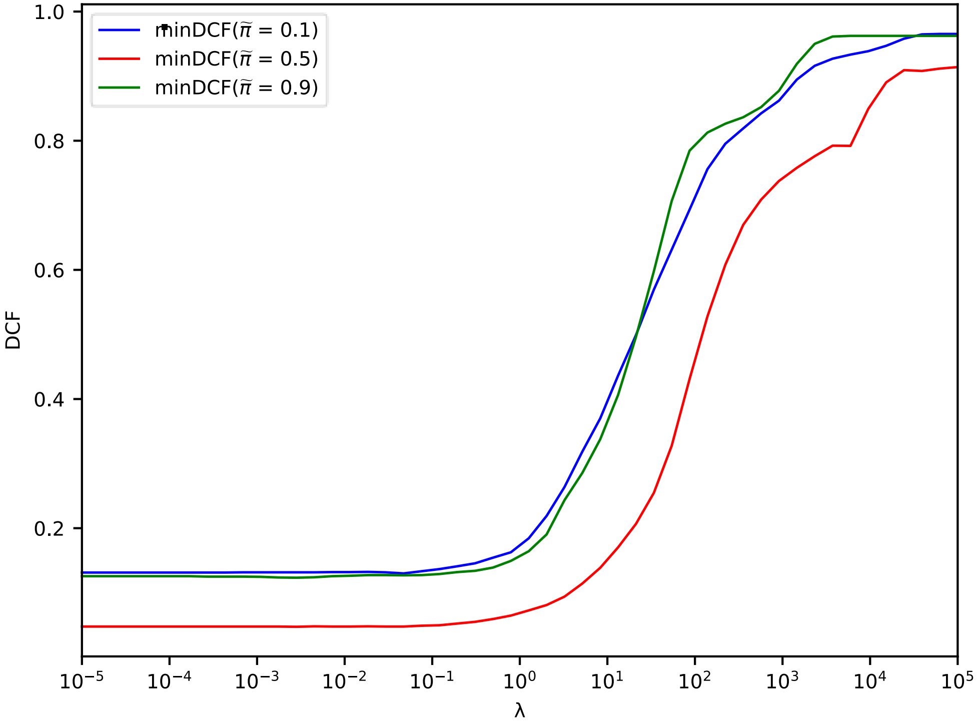
Our classes are balanced so we decided to use the traditional version which takes into account the empirical prior.

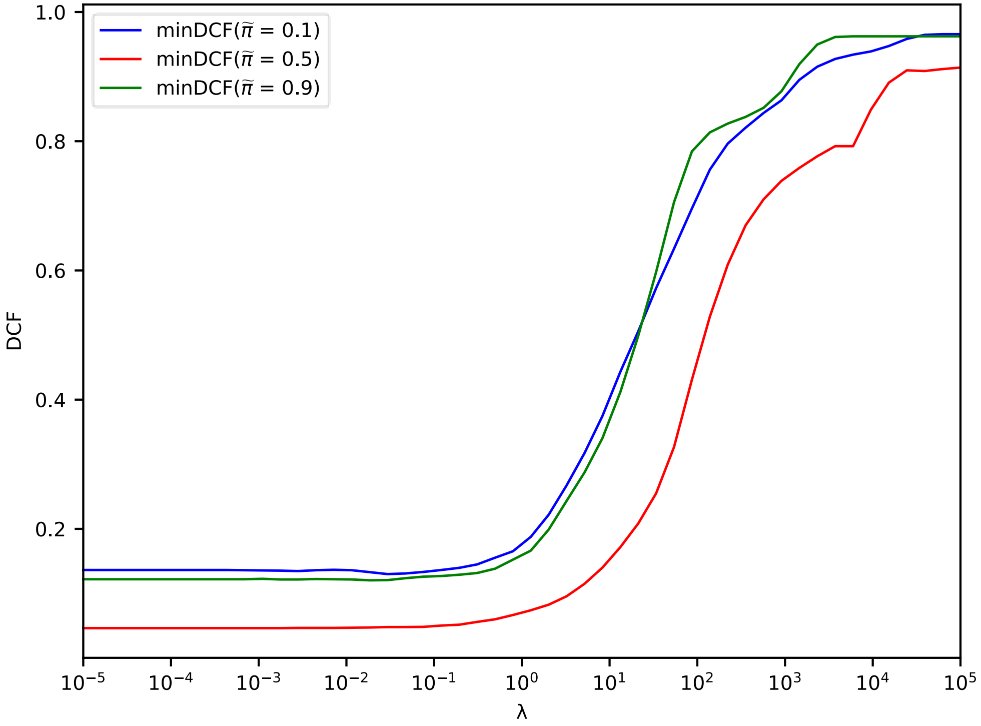
Anyway, we considered different applications (0.1, 0.5, 0.9) and compare the models using different values of λ.

Logistic Regression

Linear Log-Reg: min DCF for different values of λ with raw data.

Left: no pre-processing. Right: PCA (m = 8)





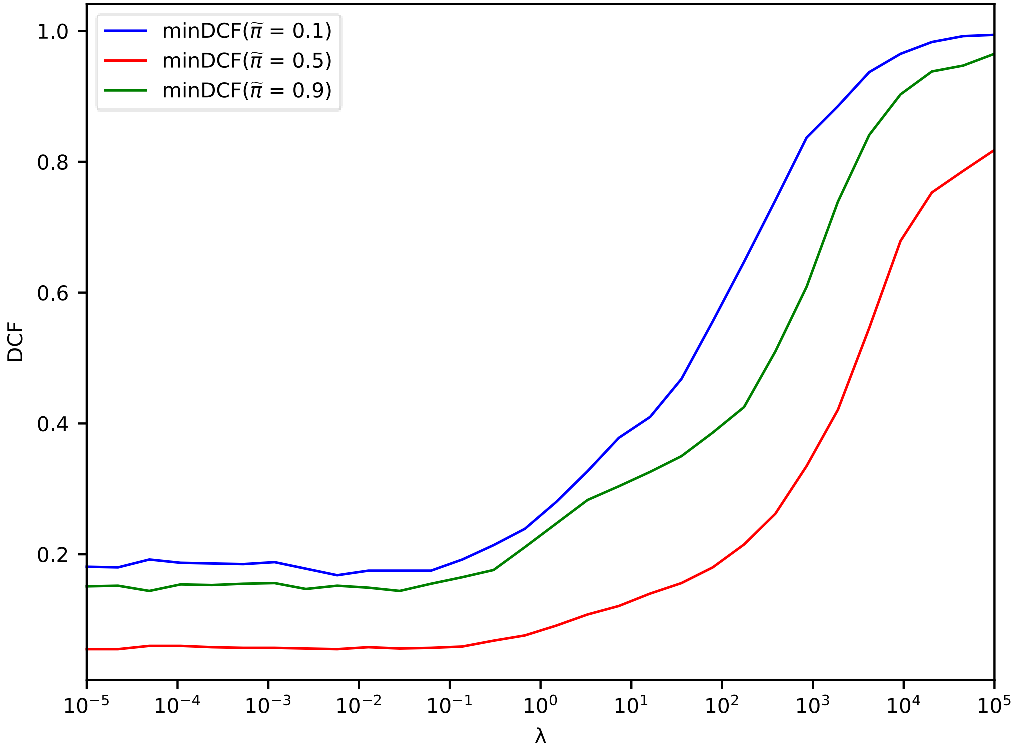
While we cannot see any differences between row data and PCA with m = 8, regularization provides a large difference. Best results are obtained with small values of λ. This means that we will get a solution that has good separation on the training set but may have poor classification on unseen data.

Given that logistic regression does not improve the performance we will not carry on this method.

Quadratic Logistic Regression

The Quadratic Log-Reg gives us worse results than the linear one, especially for the unbalanced applications.

.



As before, we consider training using a different effective prior to see the effects on the other applications. We restrict the analysis to models with small regularization.

	$\tilde{\pi}$ = 0.1	$\tilde{\pi}$ = 0.5	$\tilde{\pi}$ = 0.9
RAW DATA			
MVG (Full-Cov)	**0.126**	0.048	0.123
Log Reg (λ = 1e-05)	0.131	0.048	0.126
Quad Reg (λ = 1e-05)	0.172	0.054	0.158
PCA (m=8)			
MVG (Full-Cov)	0.136	**0.046**	**0.118**
Log Reg (λ = 1e-05)	0.136	**0.046**	0.122

We do not see great defferences between PCA with m = 8 and raw data. We can just notice a small improvement over the MVG (Full-Covariance) respect to Logistic Regression.

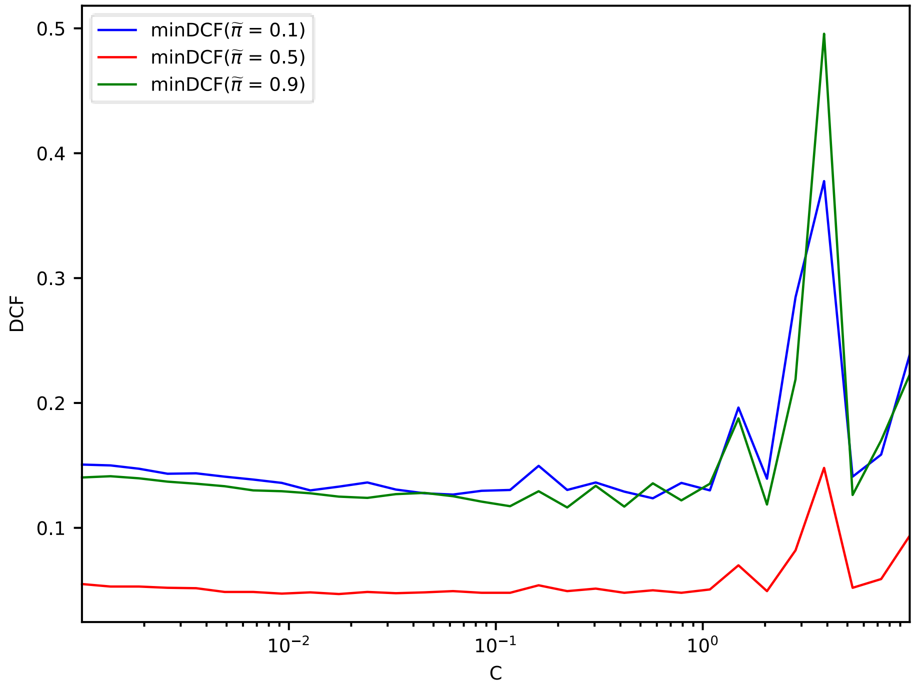
The Gaussian assumption are almost accurate for our features but, given that 4 of our features seem to be composed of more than one gaussian, we expect to have good results with Gaussian Mixture Models, which could better fit the distribution .

Anyway, we start with analyzing linear SVM.

For linear SVM, we need to tune the hyper-parameter C.

Linear SVM

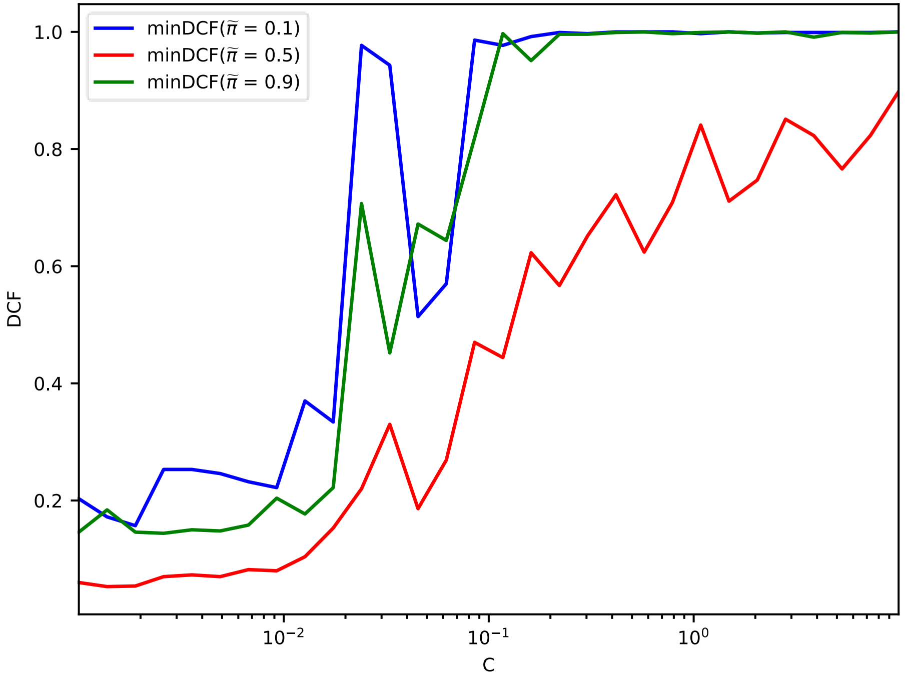
Linear SVM: min DCF for different values of C, K=5, no gaussianized features, no balancing.



In this case, the choice of C affects the performance.

Quadratic SVM

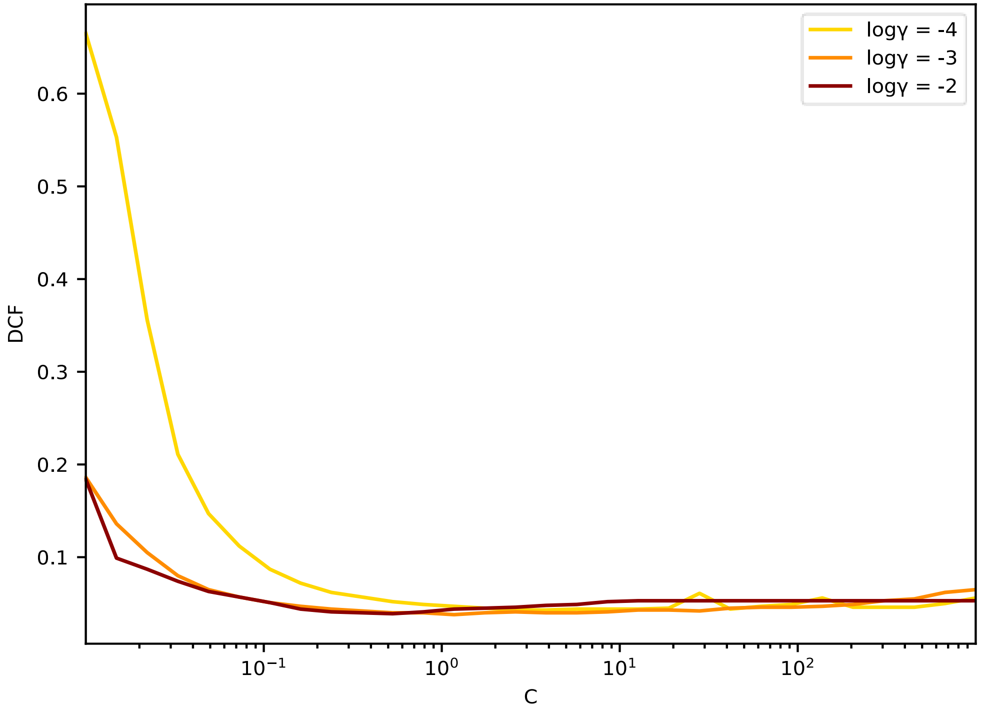
Quadratic kernel SVM: min DCF for different values of C.
Our primary metric is the red line



Also in this case, the choice of C affects the performance.
We can notice a better performance with small values of C.

RBF kernel SVM

RBF kernel SVM: min DCF with effective prior and for different values of C and γ.
No Gaussianized features and k =5.



The plot shows that both γ and C influence the results. Furthermore, both should be optimized jointly since the optimal C depends on the chosen γ. We choose log γ = -3

The best results are reported in the table below.
We can notice that we obtain the best result with the SVM classifier using RBF Kernel and the raw data.

	$\tilde{\pi}$ = 0.1	$\tilde{\pi}$ = 0.5	$\tilde{\pi}$ = 0.9
RAW DATA			
Linear SMV (C=1e-1)	0.134	0.049	0.120
Quad SVM (C=1.8e-3)	0.148	0.053	0.143
RBF SVM(log γ = -3, C=1)	0.123	0.039	0.120
PCA (m=8)			
Linear SMV (C=1e-1)	0.171	0.051	0.180
Quad SVM (C=1.8e-3)	0.136	0.046	0.122
RBF SVM(log γ = -3, C=1)	0.123	0.042	0.124

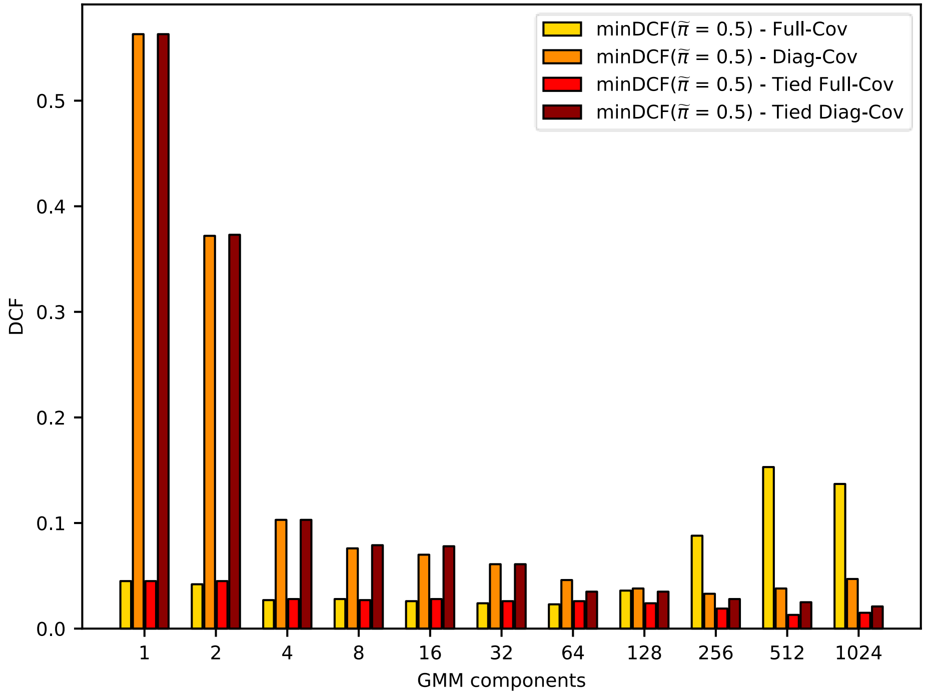
As we said, GMMs can approximate generic distributions, so we expect to obtain better results than with the Gaussian model.

We consider both full covariance and diagonal models, with and without covariance tying.

We use the K-fold with k = 5 to select the number of Gaussians and to compare different models

Gaussian Mixture Model

GMM: min DCF(effective prior = 0.5) with raw features.



In general, the performance improves with more components. Furthermore, we can notice that the Tied Full-Covariance variant performs better with all the values.

On average, the non diagonal variants perform better even if the Full-Covariance shows some degree of over-fitting when the number of components becomes large. On the contrary, the diagonal ones produce the worst results with small components. Overall, the best performance is with Tied Full-Covariance with 512 and 1024.

Despite that, we decided not using a large number of components because this could leads us to over-fitting. Furthermore, using too many components has no sense because the model estimates only the mean, since the covariance matrix is the same.

Overall, we decided to carry on the Tied Full-Covariance 512 model just because it gives the best result even if it could cause over-fitting issues.

These are the best results we obtained with GMM.

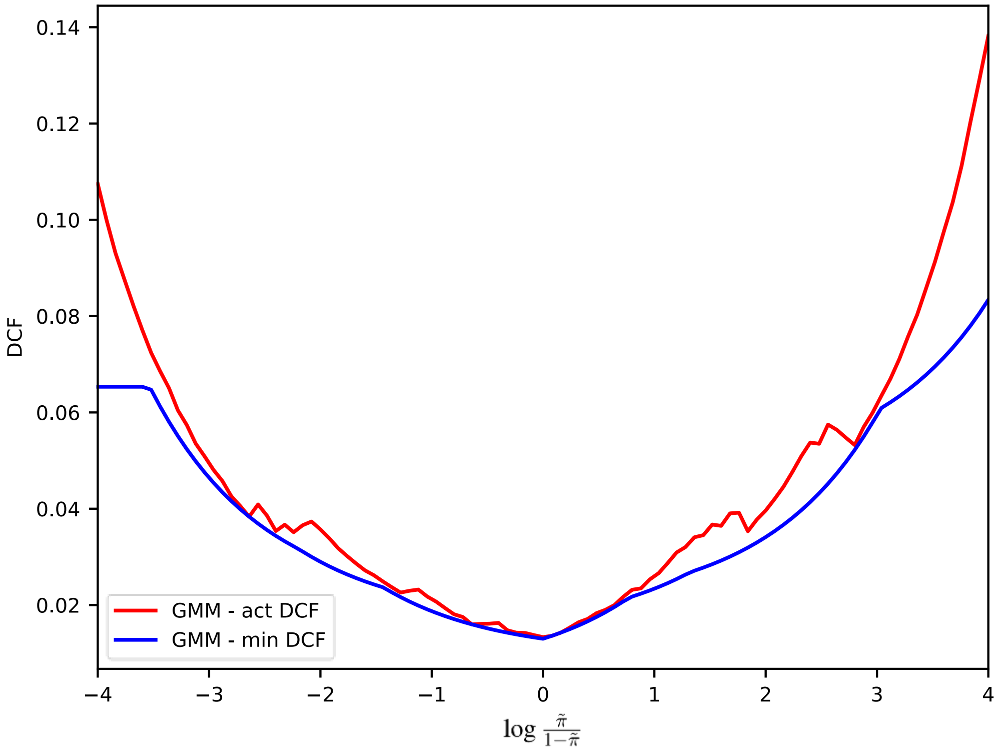
	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$
		RAW DATA	
Full cov, 64 Gau	0.081	0.023	0.060
Diag cov, 256	0.094	0.033	0.083
Tied full cov, 512	0.032	0.013	0.037
Tied diag cov, 1024	0.053	0.021	0.066
		PCA (m = 8)	
Full cov, 32 Gau	0.743	0.265	0.672
Diag cov, 16 Gau	0.550	0.229	0.632
Tied full cov, 32 Gau	0.696	0.265	0.265
Tied diag cov, 16 Gau	0.550	0.229	0.613

GMM performs consistently better than any other method in all the applications.

	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$
		RAW DATA	
Tied ful cov, 512	0.032	0.013	0.037
RBF SVM(log γ = -3, C=1)	0.123	0.039	0.120
		PCA (m=8)	
MVG (Full-Cov)	0.136	0.046	0.118
Log Reg (λ = 1e-05)	0.136	0.046	0.122

Bayes error plot

Raw data with 512 components (uncalibrated).



	$\tilde{\pi} = 0.1$		$\tilde{\pi} = 0.5$		$\tilde{\pi} = 0.9$	
	min DCF	act DCF	min DCF	act DCF	min DCF	act DCF
GMM (Full Tied-Cov)	0.032	0.034	0.013	0.013	0.037	0.046

The model provides scores that are already well-calibrated over a wide range of applications because the GMM is a generative model. In fact, the density of the two classes are independently estimated, so rebalancing does not leads to great improvements.
However we will try to calibrate the scores with the prior weighted Logistic regression model.

Calibration

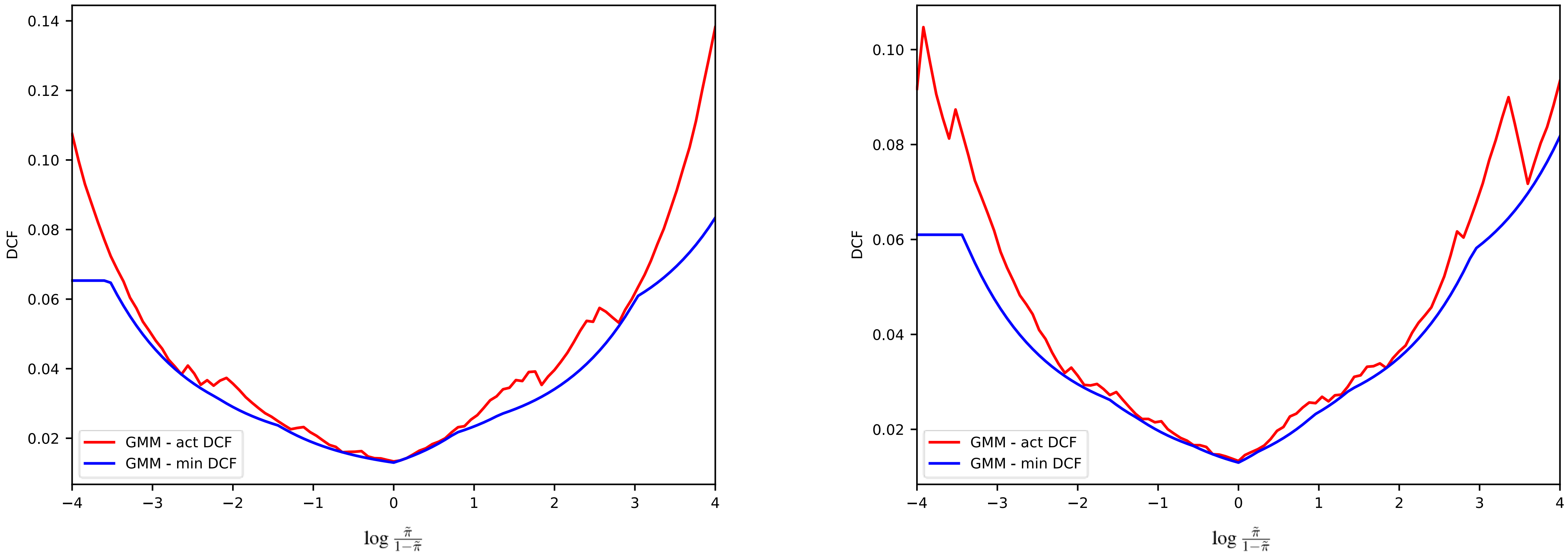
We computed the score calibration using again the k-fold approach with k = 5.

	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$
	min DCF	min DCF	min DCF
GMM (Full Tied-Cov)	0.032	0.013	0.037
	act DCF	act DCF	act DCF
Uncalibrated	0.034	0.013	0.046
Log-Reg	0.032	0.013	0.041

As expected, the calibration with prior weighted Logistic Regression approach provides slightly better results.

Bayes error plot – Calibration

left: non-calibrated scores; right: calibrated scores (model trained with $\tilde{\pi}$ = 0.5)



EXPERIMENTAL RESULTS

	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$
RAW DATA (no PCA)			
Full-Cov	0.134	0.053	0.138
Diag-Cov	0.810	0.570	0.882
Tied Full-Cov	**0.133**	**0.051**	0.135
Tied Diag-Cov	0.808	0.570	0.880
PCA (m = 8)			
Full-Cov	0.143	0.053	0.139
Diag-Cov	0.201	0.068	0.178
Tied Full-Cov	0.142	0.053	**0.134**
Tied Diag-Cov	0.202	0.068	0.180
Gaussianized features (no PCA)			
Full-Cov	0.202	0.073	0.182
Diag-Cov	0.791	0.547	0.846
Tied Full-Cov	0.184	0.069	0.177
Tied Diag-Cov	0.793	0.545	0.847

Results are consistent with our expectations. In this case, the best model is the Tied Full Covariance, while before we have considered the full Covariance.
PCA with m = 8 does not provide good improvements as for the validation set.
Gaussianization supplies results similar to the previous ones.

EXPERIMENTAL RESULTS

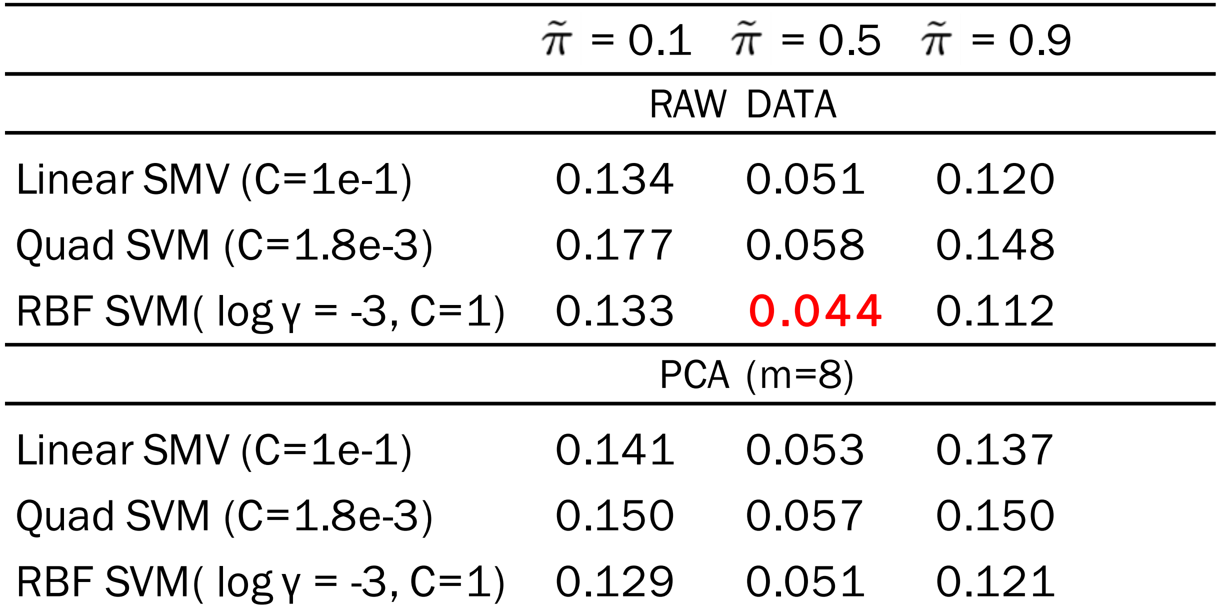
Logistic Regression

	$\tilde{\pi}$ = 0.1	$\tilde{\pi}$ = 0.5	$\tilde{\pi}$ = 0.9
RAW DATA			
Quad Reg (λ = 1e-05)	0.202	0.063	0.152
Log Reg (λ = 1e-05)	0.135	0.053	0.133
PCA (m=8)			
Log Reg (λ = 1e-05)	0.144	**0.052**	0.136

With respect to the validation phase, these results are slitghly worse.

EXPERIMENTAL RESULTS

SVM



	$\tilde{\pi}$ = 0.1	$\tilde{\pi}$ = 0.5	$\tilde{\pi}$ = 0.9
	RAW DATA		
Linear SMV (C=1e-1)	0.134	0.051	0.120
Quad SVM (C=1.8e-3)	0.177	0.058	0.148
RBF SVM(log γ = -3, C=1)	0.133	0.044	0.112
	PCA (m=8)		
Linear SMV (C=1e-1)	0.141	0.053	0.137
Quad SVM (C=1.8e-3)	0.150	0.057	0.150
RBF SVM(log γ = -3, C=1)	0.129	0.051	0.121

As in the validation phase, the best result is provided by the RBF SVM. Also the other results are similar with what we obtain before.

EXPERIMENTAL RESULTS

GMM

	$\tilde{\pi}$ = 0.5
	RAW DATA
Full Cov, 8 Gau	0.032
Diag Cov, 32 Gau	0.079
Tied Full-Cov, 4 Gau	**0.029**
Tied Diag-Cov, 128 Gau	0.073

Tied Full-Cov, 512 Gau	0.054

These are the best results with GMM.
The same model we chose during the validation phase (the last line) is worse in this case; this could be related, as we expected, to the number of components chosen, which is evidently too large for our task.