Machine Learning & Pattern Recognition

A.A. 2021/2022

Project – Gender Detection

Jacopo De Cristofaro
s302298

Abstract

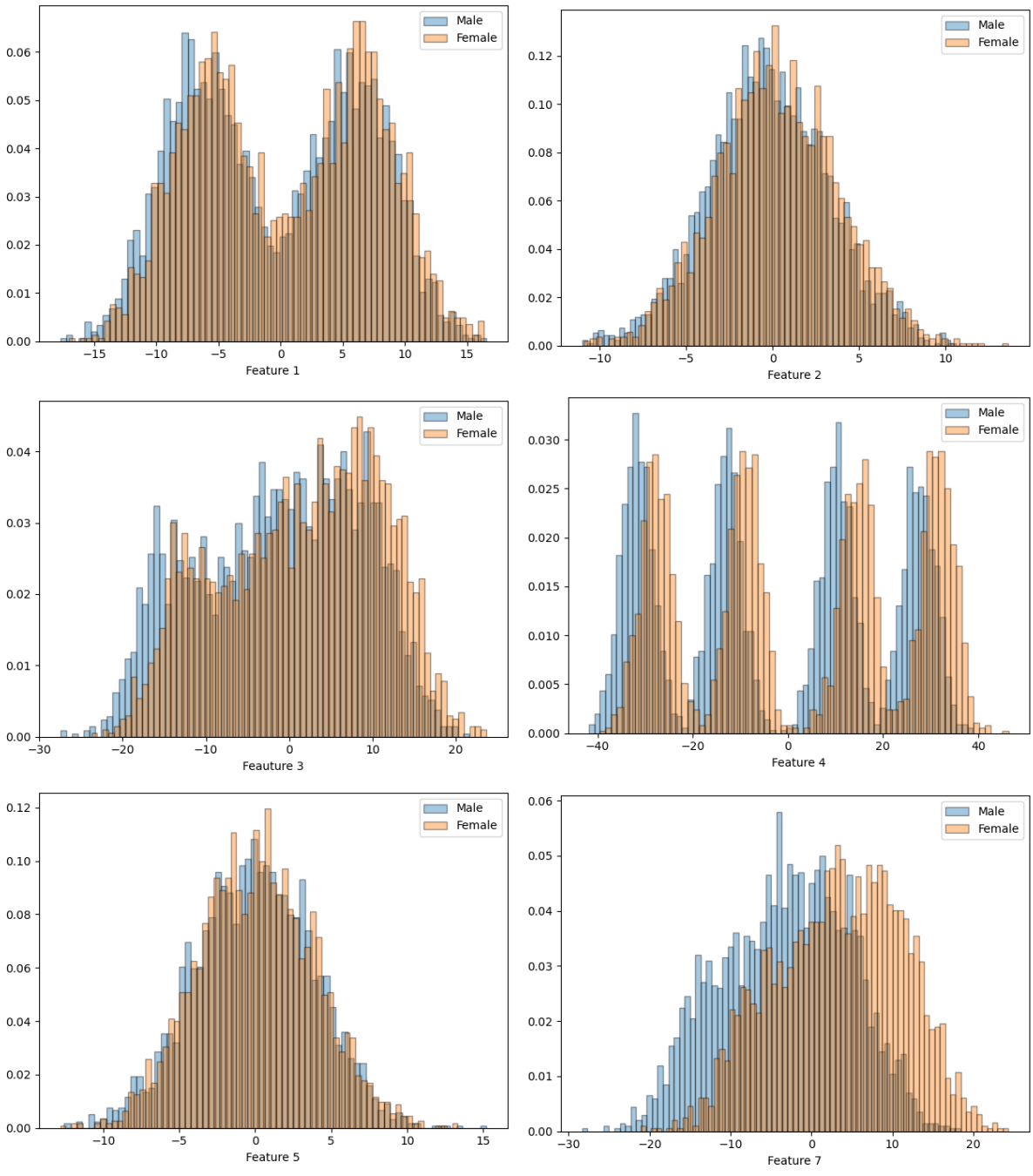
The goal of this report is to analyse a dataset composed by samples of male and female embeddings using Machine Learning tools. We will start analysing the distributions and correlations between features and then focusing on evaluating the performances of various models employing also different pre-processing techniques.

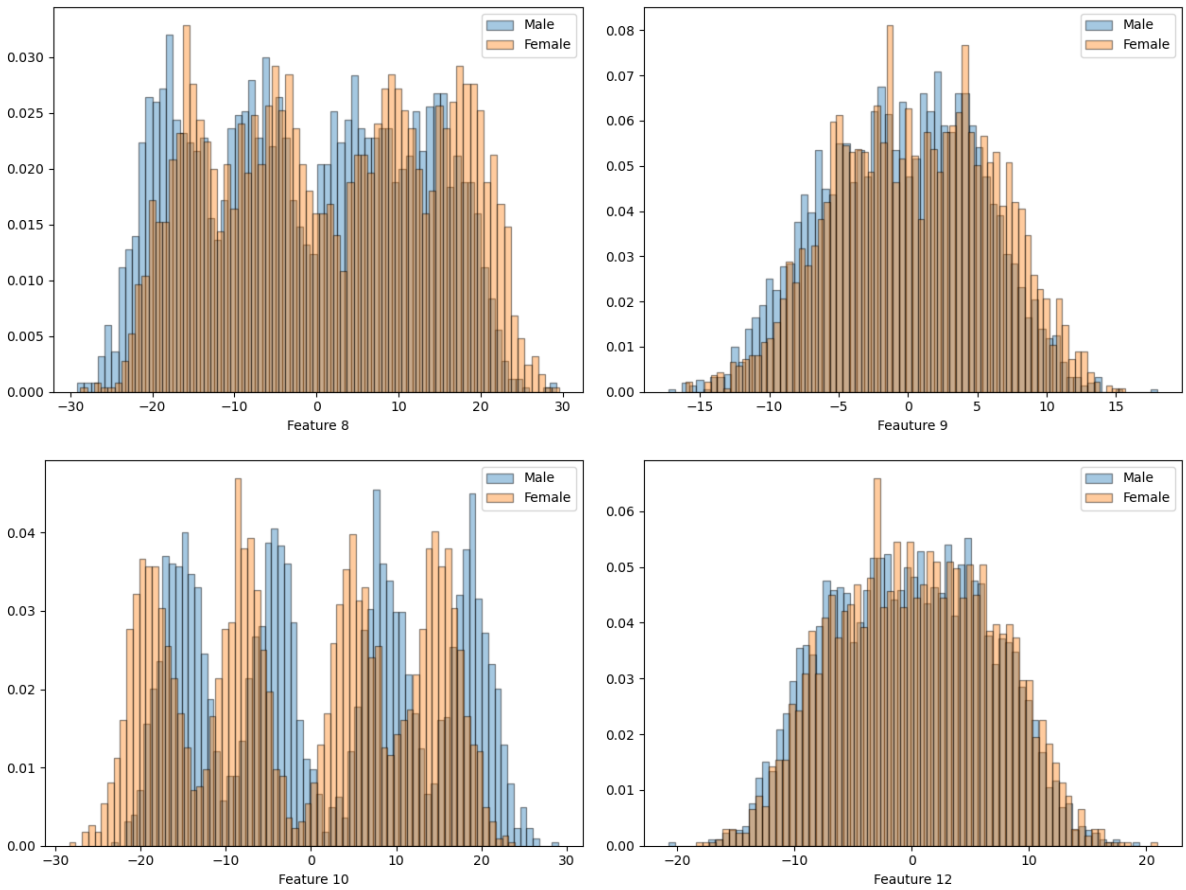
1 Application information

The training dataset is composed by 6000 samples of male and female speaker embeddings. Each embedding is a small-dimensional fixed sized representation of an utterance. The 12 features are continuous values that represent a point in the m dimensional embedding space and they do not have any interpretation. These sample belong to 4 different age groups, but the age information is not available. The dataset is balanced, there are 3000 male samples and 300 female ones. The test dataset is composed by 2000 male samples and 2000 female ones.

1.1 Dataset features

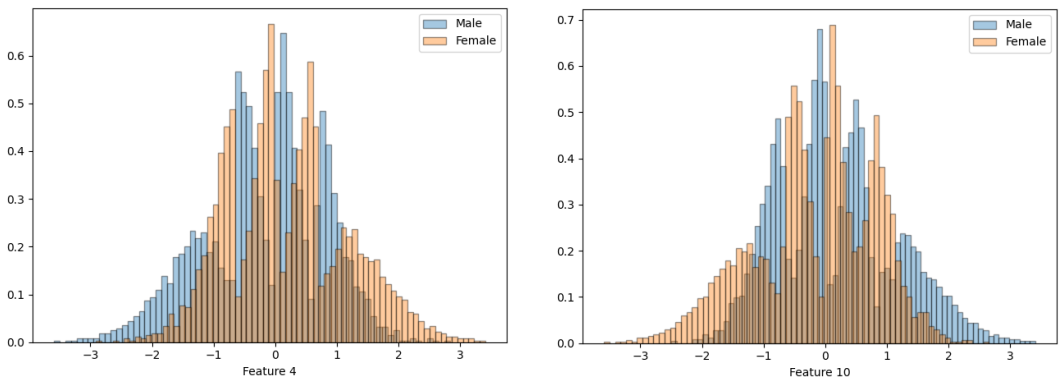
The following histograms show how the 12 features, which were previously centred, are distributed:





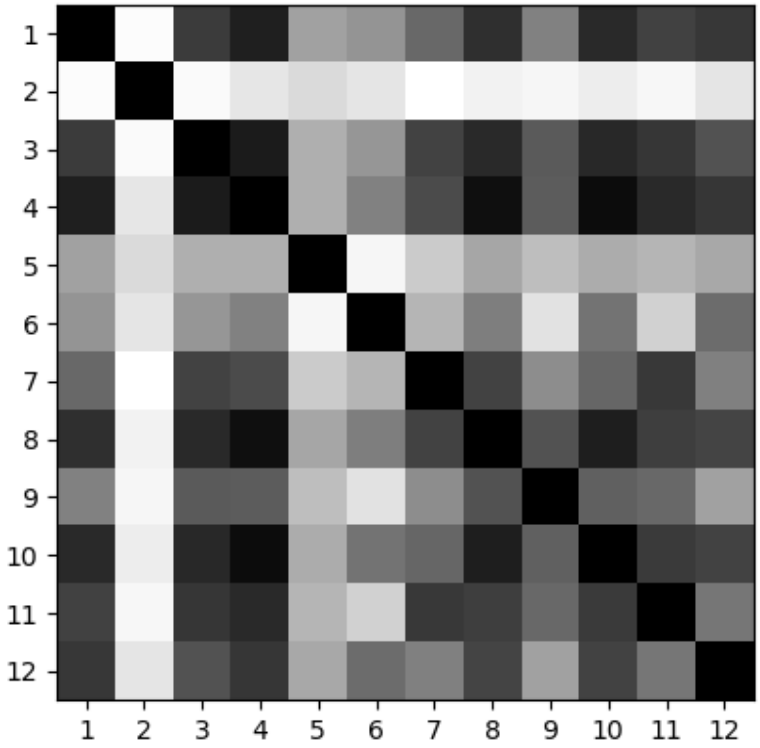
The within class distribution is basically the same for all the features. It is interesting to notice that several features like the 4, 8, 10 are made up by 4 cluster normally distributed, most likely each cluster belongs to one of the age groups discussed before. Other features like 2, 5, 6 are distributed normally while the remaining one do not have any well-known shape (some of them recall a gaussian one). The features that will be contribute more are 4, 6, 7, 10 since the male and the female histograms do not significantly overlap like the other ones. All of them have comparable range scales, so most likely performance for raw features and the Z-Scored ones will be very similar.

To “reshape” the distribution of the features into a gaussian one, and to reduce the impact of the (few) outliers, we will consider the “Gaussianization” pre-processing. This transformation also has the effect to normalize the values, so all the transformed features will have the same scale. Since most of the “gaussianized” features show within class gaussian distribution, we will report only the histograms of features 4 and 10:



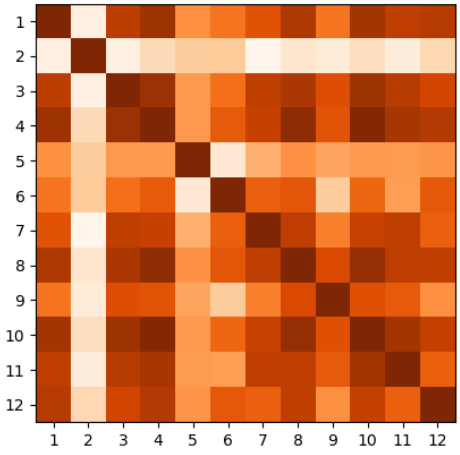
These transformed features unlike the other ones have only a global gaussian distribution. Indeed, Gaussianization is not a “supervised” technique.

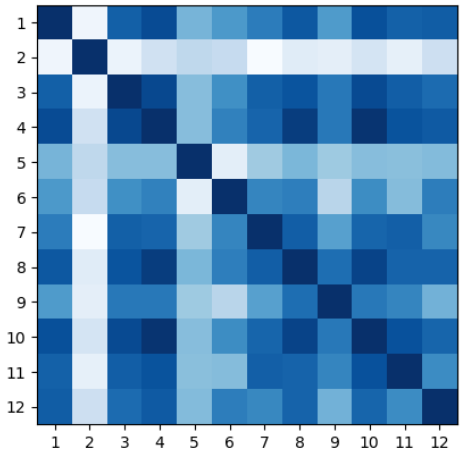
We’ll now turn our attention analysing how the raw features are correlated between them, plotting heatmaps of the absolute value of the Pearson correlation coefficient, firstly for the whole dataset and then for the two classes:



Some considerations: only the feature 1 is totally uncorrelated from the other ones. The remaining ones are moderately correlated (for most of them the coefficient is around 0.65). On the other hand, the following couples of features (3,7) and (3,9) are highly correlated (coefficient higher than 0.8), so this suggests we may benefit from using PCA to map data from 12 to 10, uncorrelated features to reduce the number of parameters to estimate.

Let’s now analyse the within-class heatmaps (left female, right male):





The heatmap are very similar to each other, so it suggests that Full Covariance Multivariate model and the Tied one will have similar performance, while the Naïve Bayes assumption will lead to scare results, since for both classes the feature are moderately correlated.

Heatmaps of the gaussianized data and the Z-Scored one will be not reported since they do not differ in a significant way from the discussed ones, so the previously discussed observations hold also for them.

1.2 Validation Methodology

To understand which model is most promising, and to assess the effects of the different pre-processing techniques, we will employ the K-Fold cross validation with $K = 5$. Using this methodology, we will have more data for training and validation (which is nice for hyperparameters tuning) and the training of the chosen model will be on the whole training dataset. Since the dataset is balanced our main application will be a uniform prior one, so:

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

We will consider also highly unbalanced scenarios like:

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

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

The goal of the following analysis is to choose the model which performances promise to be the best. So, we will measure these in terms of normalized minimum detection costs.

2 Classifiers performances

2.1 Gaussian models

In the first instance we will analyse how behave Gaussian classifiers. Also, we will try to combine different pre-process techniques like Gaussianization, Z-Score and PCA, to determine if them are effective. In the following table are reported the normalized minimum detection costs for the different scenarios:

Gaussianization	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Full Covariance	0.061	0.184	0.173
Full-Tied Covariance	0.061	0.180	0.169
Diagonal Covariance	0.540	0.806	0.826
Tied-Diagonal Covariance	0.538	0.800	0.821

Gaussianization – PCA (m = 11)	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Full Covariance	0.074	0.216	0.206
Full-Tied Covariance	0.072	0.208	0.199
Diagonal Covariance	0.086	0.229	0.227
Tied-Diagonal Covariance	0.083	0.230	0.220

Gaussianization – PCA (m = 10)	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Full Covariance	0.072	0.205	0.203
Full-Tied Covariance	0.071	0.204	0.201
Diagonal Covariance	0.084	0.232	0.224
Tied-Diagonal Covariance	0.082	0.226	0.220

Gaussianization – PCA (m = 9)	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Full Covariance	0.088	0.240	0.234
Full-Tied Covariance	0.087	0.235	0.231
Diagonal Covariance	0.094	0.256	0.258
Tied-Diagonal Covariance	0.095	0.256	0.257

Z-Score	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Full Covariance	0.048	0.129	0.125
Full-Tied Covariance	0.047	0.124	0.125
Diagonal Covariance	0.565	0.817	0.858
Tied-Diagonal Covariance	0.565	0.819	0.854

Z-Score – PCA (m = 11)	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Full Covariance	0.096	0.268	0.225
Full-Tied Covariance	0.095	0.263	0.221
Diagonal Covariance	0.105	0.271	0.243
Tied-Diagonal Covariance	0.106	0.266	0.236

Z-Score – PCA (m = 10)	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Full Covariance	0.113	0.304	0.252
Full-Tied Covariance	0.111	0.303	0.258
Diagonal Covariance	0.120	0.303	0.270
Tied-Diagonal Covariance	0.118	0.301	0.266

Z-Score – PCA (m = 9)	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Full Covariance	0.160	0.402	0.389
Full-Tied Covariance	0.159	0.404	0.387
Diagonal Covariance	0.164	0.410	0.389
Tied-Diagonal Covariance	0.162	0.414	0.383

The Full and the Tied Full Covariance models obtain the best results. The fact that the Full and Tied model are very similar (in terms of performances) may suggest that the log like-hood ratio of the scores has a linear formulation; indeed, the class heatmaps showed before suggests that the even the covariance matrices are very similar and since the decision function of the Full model is:

$$s(x) = x^T A x + x^T b + c$$

where $A = -\frac{1}{2}(\Sigma_1 - \Sigma_0)$ with $\Sigma_1$and Σ_0 being the class covariance matrices. If they are very similar, implies that $|A| \approx 0$. Plugging in this result we have:

$$s(x) \approx x^T b + c$$

Which is the Tied decision function.

PCA is not effective neither for Full nor Tied covariance model, it worse the performances. This means that the model is able to get robust estimates even with all the 12 dimensions.

As we expected the gaussian models based on the Naïve assumption, like the Diagonal and the Tied Diagonal ones, performs very bad compared to the other, but pre-processing the data also with PCA significantly improve the performance of them. Indeed, PCA can give in output transformed data with transformed features that are uncorrelated (let's remember that two uncorrelated gaussian R.V. are also independent, so empirically we can say that the within-class covariance matrices will be substantially diagonal. What will be certainly diagonal is the covariance matrix computed over all the training sample), however as we can see for Full covariance models, we are losing information. For this reason and since the number of samples of the dataset is much higher than the number of the features, PCA will not be considered anymore.

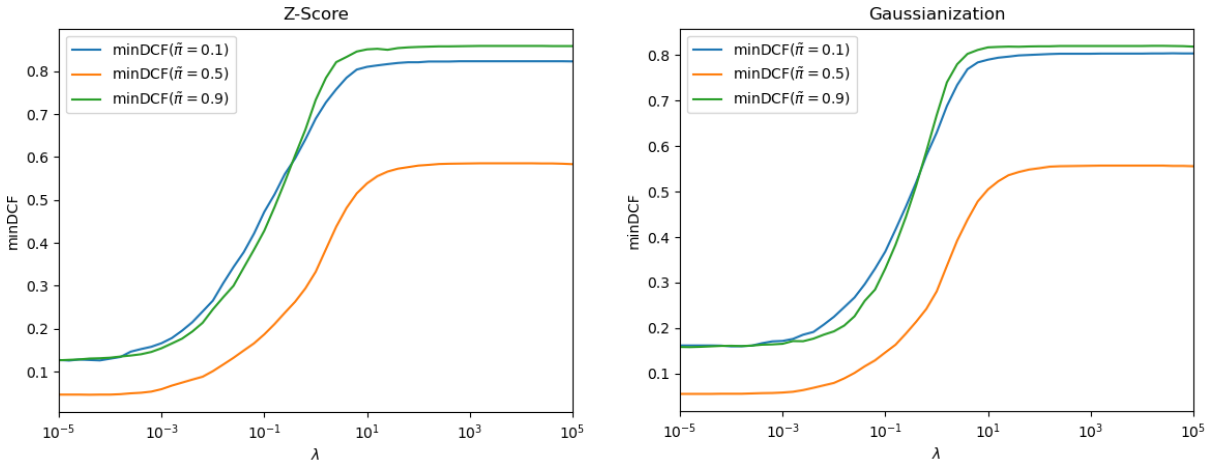
Gaussianization pre-processing seems to be not very helpful, the best results are obtained with Z-Scored data. On the other hand, gaussianized data are affected less than the other ones by PCA.

2.2 Logistic regression models

Now we will focus on the regularized linear Logistic Regression model, using the following prior-weighted formulation of the objective function, in order to try to optimize the classifier also for the secondary applications (NOTE: for $\pi_t = 0.5$, in our study case, most likely will corresponds to the non-prior weighted version since each fold is approximately balanced, for this reason will be considered only the prior-weighted version of the classifier).

$$J(\boldsymbol{w},b)=\frac{\lambda}{2}\|\boldsymbol{w}\|^2+\frac{\pi_T}{n_T}\sum_{i=1|c_i=1}^{n}\log\left(1+e^{-z_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)}\right)+\frac{1-\pi_T}{n_F}\sum_{i=1|c_i=0}^{n}\log\left(1+e^{-z_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)}\right)$$

The Logistic Regression model does not make any assumptions on the distribution of the data, so we can expect that the Gaussianization pre-processing will not have much effect. Let's see how the model behave tuning the hyper-parameter λ from 10^{-5} to 10^{5} in a logarithmic fashion (was calculated also the performance for $\lambda = 0$) setting $\pi_t = 0.5$.



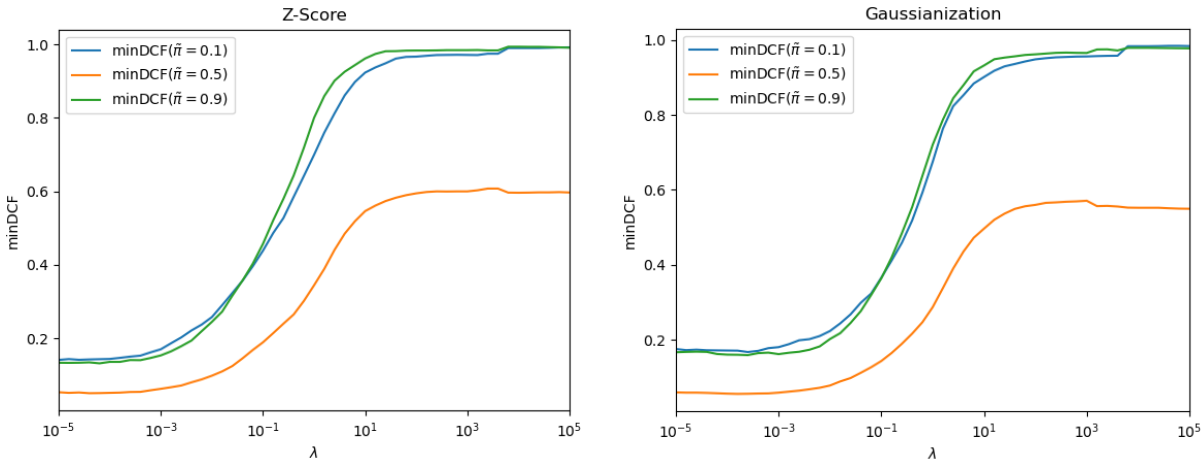
For values of λ too large, we can observe that the model is not able to correctly classify the samples, since as λ increase we are introducing too much "generalization" (the model is too much simple). Meanwhile for small values we get a solution that has a good separation on the training data and for unseen data. The regularization term provides no benefit, for this reason will report only the value of the non-regularized version of the model (is better to select a value of $\lambda = 10^{-4}$ since performances are the same but minimize over-fitting).

Z-Score	$\tilde{\pi}=0.5$	$\tilde{\pi}=0.1$	$\tilde{\pi}=0.9$
Logistic Regression ($\pi_t = 0.5,\ \lambda = 0$)	0.047	0.127	0.126
Logistic Regression ($\pi_t = 0.1,\ \lambda = 0$)	0.047	0.134	0.128
Logistic Regression ($\pi_t = 0.9,\ \lambda = 0$)	0.047	0.129	0.127

Gaussianization	$\tilde{\pi}=0.5$	$\tilde{\pi}=0.1$	$\tilde{\pi}=0.9$
Logistic Regression ($\pi_t = 0.5,\ \lambda = 0$)	0.056	0.162	0.159
Logistic Regression ($\pi_t = 0.1,\ \lambda = 0$)	0.055	0.172	0.166
Logistic Regression ($\pi_t = 0.9,\ \lambda = 0$)	0.057	0.164	0.158

Since Z-Score is a linear transformation, the performances obtained on the raw data (not reported here) and the Z-Scored data are the same; this means that Gaussianization does not provide any benefit, actually the performances worse. The optimized versions of the classifier for the two secondary applications cannot achieve better results than the main one (the model trained with $\pi_t = 0.5$).

We will now focus on the Quadratic version of the model and as we did for the Linear version, we will tune λ in the same way.



The observation on the value of λ done before holds also in this case for Z-Scored data, meanwhile for the gaussianized one with $\lambda = 10^{-4}$ the model performs slightly better.

Z-Score	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Quadratic LR ($\pi_t = 0.5,\ \lambda = 0$)	0.053	0.141	0.133
Quadratic LR ($\pi_t = 0.1,\ \lambda = 0$)	0.055	0.153	0.147
Quadratic LR ($\pi_t = 0.9,\ \lambda = 0$)	0.053	0.155	0.140

Gaussianization	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Quadratic LR ($\pi_t = 0.5,\ \lambda = 10^{-4}$)	0.057	0.172	0.160
Quadratic LR ($\pi_t = 0.1,\ \lambda = 10^{-4}$)	0.053	0.166	0.172
Quadratic LR ($\pi_t = 0.9,\ \lambda = 10^{-4}$)	0.060	0.188	0.164

The overall performance of the model is little worse than the Linear counterpart. This suggests that linear separation rule fit better than quadratic ones for this task, so for the moment the best candidate to be chosen as final classifiers are the Full Covariance Gaussian Classifier, also the Tied version and the Linear Logistic Regression one.

2.3 Support Vector Machine model

The next models that we will employ are the Support Vector Machine models. We will start our analysis with the linear version and later using different kernels. As we did for the Regression model, we will consider the prior-weighted version of the classifier. To re-balance the class we use a different value of C for the different classes:

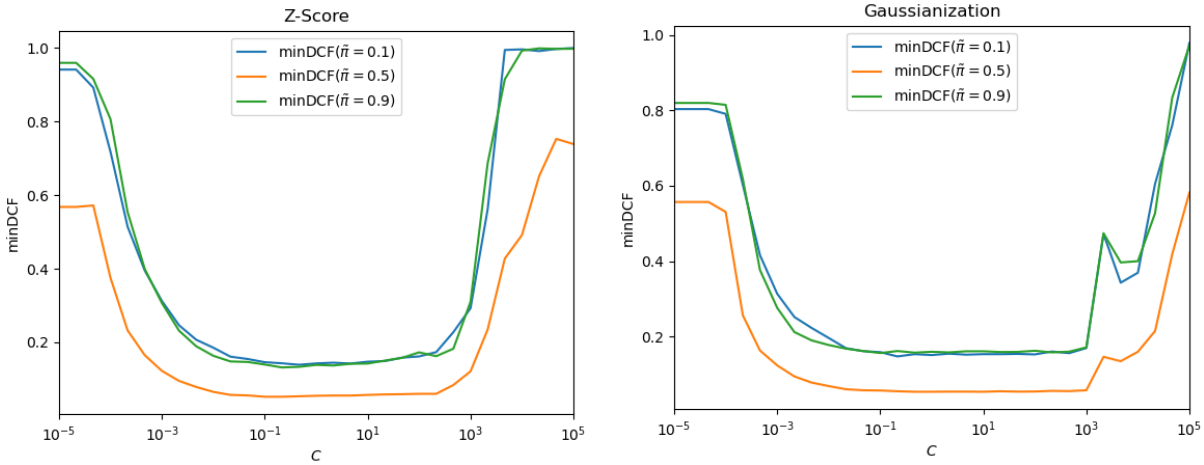
$$\max_{\alpha} \boldsymbol{\alpha}^T \mathbf{1} - \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{H} \boldsymbol{\alpha}$$

subject to:

$$0 \leq \alpha_i \leq C_i,\ i = 1, \ldots, n$$

where $C_i = C\frac{\pi_T}{\pi_T^{emp}}$ for samples of the male class, while $C_i = C\frac{\pi_F}{\pi_F^{emp}}$ for samples of the female one. π_T^{emp} and π_F^{emp} are the empirical priors for the two classes computed over the training data.

Previous results suggests that linear decision rules can well separate classes, so we expect to get good behaviour from the linear formulation of the model. It is characterized by the hyper-parameter C, so it will be tuned from 10^{-5} to 10^{5} in a logarithmic fashion.



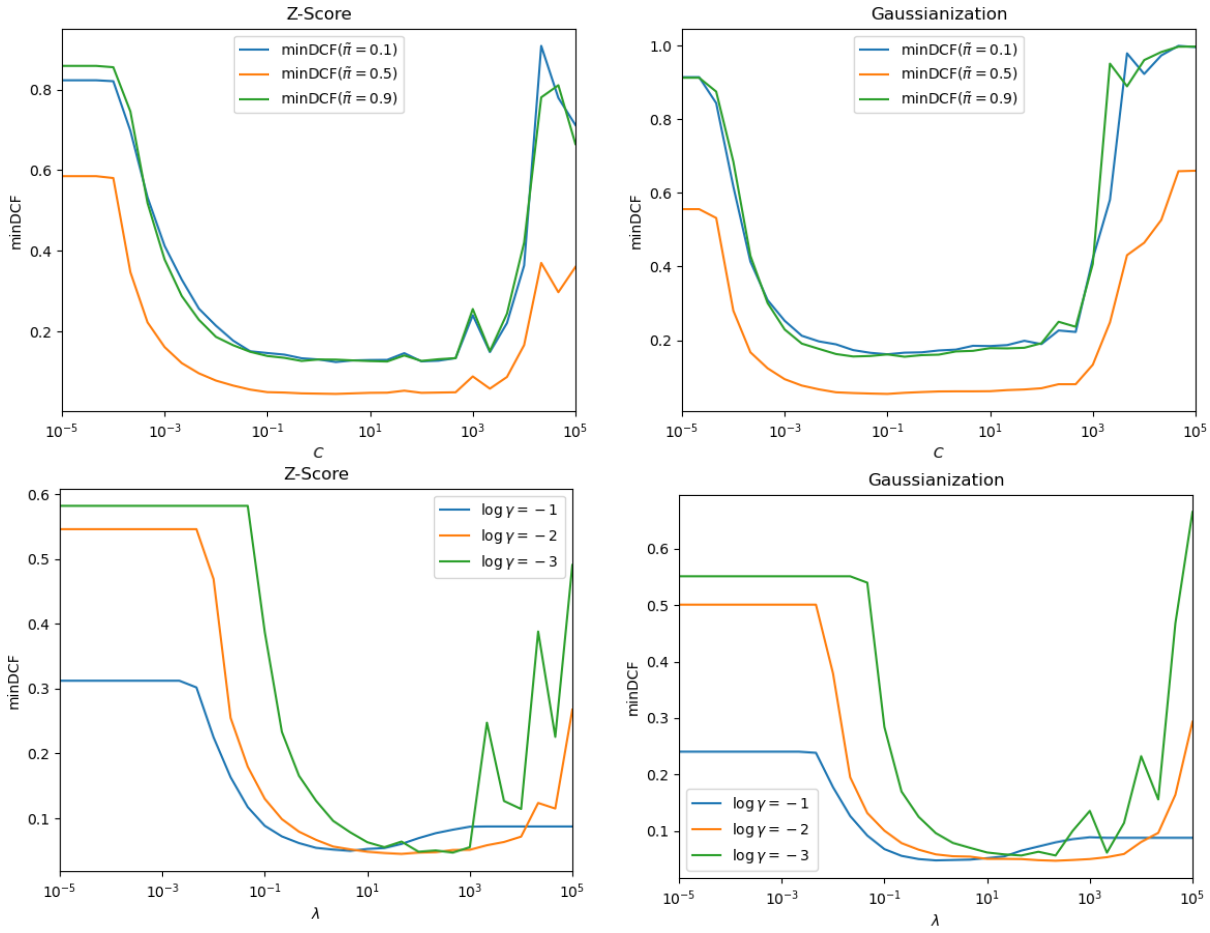
The choice of C looks critical for both version of the data. The value of C to chose is 1, which optimize all the different applications that we are interested in.

Z-Score	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Linear SVM ($\pi_t = 0.5, C = 1$)	0.046	0.131	0.131
Linear SVM ($\pi_t = 0.1, C = 1$)	0.048	0.136	0.131
Linear SVM ($\pi_t = 0.9, C = 100$)	0.046	0.131	0.127

Gaussianization	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Linear SVM ($\pi_t = 0.5, C = 1$)	0.054	0.151	0.160
Linear SVM ($\pi_t = 0.1, C = 10$)	0.055	0.165	0.160
Linear SVM ($\pi_t = 0.9, C = 1000$)	0.062	0.061	0.157

As we can see also for this linear model the obtained results are good and about the same of the previous examined linear classifier, so the assumption that this kind of model works well is reinforced. Gaussianization pre-processing overall is not helpful in comparison with Z-Score, but the specialized trained model with $\pi_t = 0.9$ surprisingly obtains the best results for the secondary application with $\tilde{\pi} = 0.1$ among all the so-far trained classifier. However, since the performances on the primary application are quite poor it will be discarded.

Let's turn our attention to SVMs that exploit kernel which generate more complex separation surfaces like the Quadratic (top) and the Radial Basis (bottom). For both classifiers will be tuned the hyper-parameter C, for the Radial Basis one will be tuned jointly with γ



Even in this case the choice of C Is extremely important: for the Polynomial SVM extreme values of C, lead the classifier to act very bad. The previous statement holds also for the Radial Basis kernel SVM (the model trained with $\gamma = 0.001$, actually, shows a good overall performance) and for all the three values of γ there is a common "behavioural" pattern.

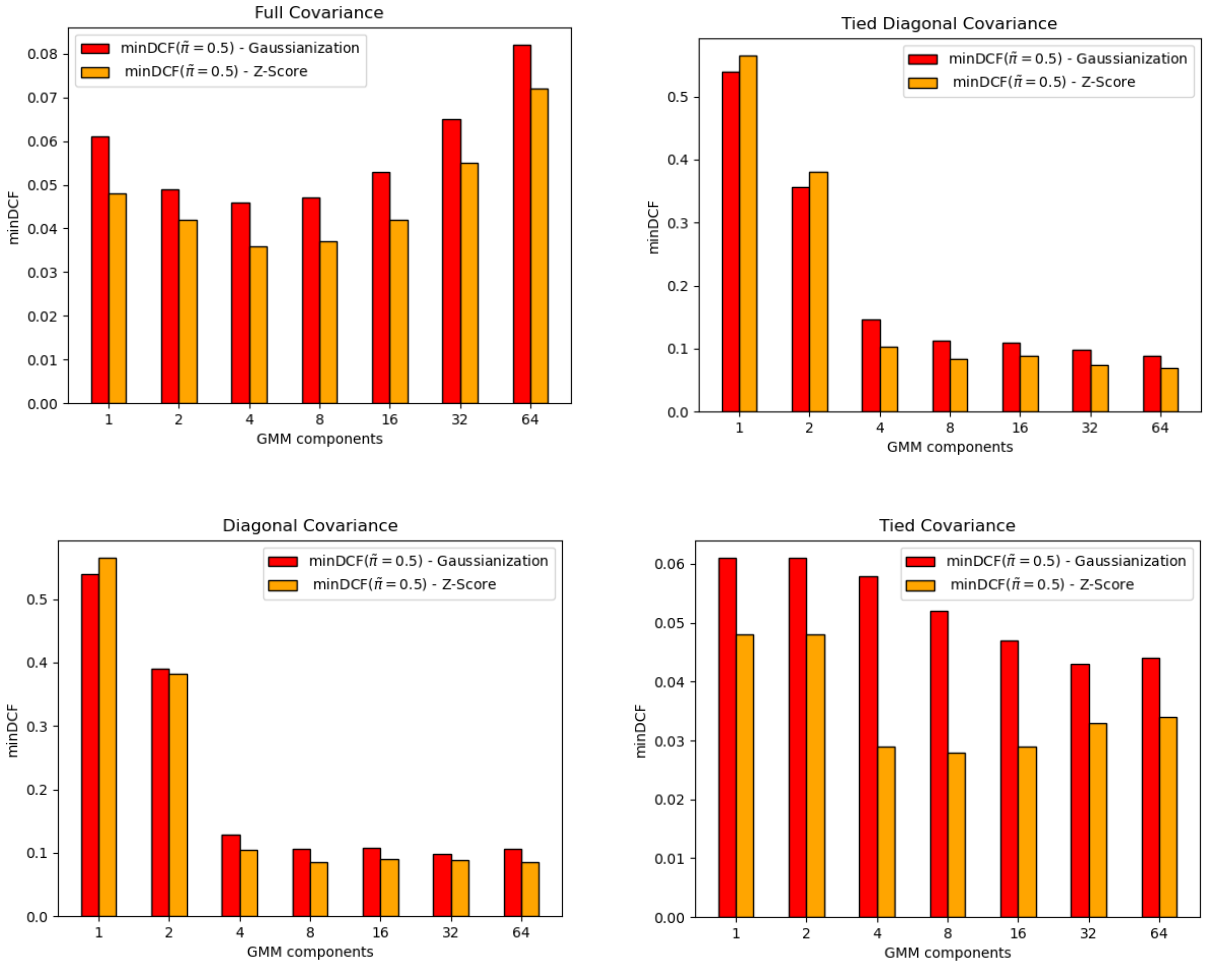
Z-Score	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Quadratic SVM ($\pi_t = 0.5, C = 0.1$)	0.052	0.146	0.140
Quadratic SVM ($\pi_t = 0.1, C = 1.0$)	0.057	0.154	0.168
Quadratic SVM ($\pi_t = 0.9, C = 1.0$)	0.055	0.155	0.138
RBF SVM ($\pi_t = 0.5, C = 100, \gamma = 0.01$)	0.047	0.134	0.129
RBF SVM ($\pi_t = 0.1, C = 10, \gamma = 0.1$)	0.087	0.159	0.162
RBF SVM ($\pi_t = 0.9, C = 10, \gamma = 0.1$)	0.087	0.179	0.148

Gaussianization	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Quadratic SVM ($\pi_t = 0.5, C = 0.1$)	0.054	0.162	0.162
Quadratic SVM ($\pi_t = 0.1, C = 0.1$)	0.059	0.166	0.197
Quadratic SVM ($\pi_t = 0.9, C = 1.0$)	0.066	0.196	0.162
RBF SVM ($\pi_t = 0.5, C = 1, \gamma = 0.1$)	0.048	0.142	0.125
RBF SVM ($\pi_t = 0.1, C = 1, \gamma = 0.1$)	0.087	0.155	0.148
RBF SVM ($\pi_t = 0.9, C = 1, \gamma = 0.1$)	0.087	0.153	0.141

Quadratic SVM obtains the same results of the Quadratic Logistic Regression, which are not good as the linear version of the two mentioned models. However, a Radial Basis kernel SVM is able to act good as the best models found so far for both Z-Scored data and Gaussianized ones (while the other ones are very good only on the former).

2.4 Gaussian Mixture models

The last classifier family that we will employ is the Gaussian Mixture one. It will be considered the Full Covariance and the Diagonal Covariance variants, with and without the tied assumption (on the components of the same class). In this case the hyperparameter to tune is the number of the Gaussian components of the model.



As we expected the two Diagonal models produce not good results since for both classes the features are moderately correlated as has been shown in the above heatmaps. The Full Covariance model trained on the Z-Score behave very well; a result that it could have been predicted in advance since, as we can see from the histograms discussed previously and from the application information (the samples belong to 4 different age groups) it was reasonable to think that the sample can be divided easily into clusters. Gaussianization even for this family is not helpful.

Z-Score	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Full Covariance (4 comp)	0.036	0.106	0.105
Full-Tied Covariance (4 comp)	0.029	0.096	0.075
Diagonal Covariance (8 comp)	0.086	0.204	0.199
Tied-Diagonal Covariance (64 comp)	0.068	0.184	0.192

Gaussianization	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Full Covariance (4 comp)	0.046	0.139	0.123
Full-Tied Covariance (32 comp)	0.043	0.123	0.125
Diagonal Covariance (32 comp)	0.098	0.263	0.272
Tied-Diagonal Covariance (64 comp)	0.088	0.280	0.265

The Full-Tied Covariance model with 4 components trained on Z-Scored data is able to generate the best results among all of the other previously analysed models, reducing by a fair amount the minimum detection cost for all the three applications. Also, the standard Full Covariance model with 4 components can lead to results that are better than all the other previously discussed classifiers. The reason why the Tied assumption helps in this case is due, most likely, to the fact that the estimates of the two within class covariance (which each of them is shared among the Gaussian component of the classes) are more robust than the one obtained by not considering it (also, the clusters found before, during the features analysing, have substantially the same shape, differing only by their mean).

The Full-Tied Covariance model with 8 components trained on Z-Scored data is very slightly better than the one with 4 components. However, we know from the application information that the samples belong to 4 different age groups, so we will choose 4 as number of components since it follows better how the data are naturally clustered.

The best classifiers at this point are, from each family:

Z-Score	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
MVG (Full-Tied Covariance)	0.047	0.124	0.125
Linear LR ($\pi_t = 0.5,\ \lambda = 0$)	0.047	0.127	0.126
Linear SVM ($\pi_t = 0.5, C = 1$)	0.046	0.131	0.131
RBF SVM ($\pi_t = 0.5, C = 100, \gamma = 0.01$)	0.047	0.134	0.129
Full Covariance (4 comp)	0.036	0.106	0.105
GMM (Full-Tied Covariance - 4 comp)	0.029	0.097	0.077

Since the gap between the GMM model and the other one is very large, for all the three considered application, we considered it as the main candidate for the evaluation phase. However, we need to check if the scores of this model need to be calibrated or not.

3 Score calibration

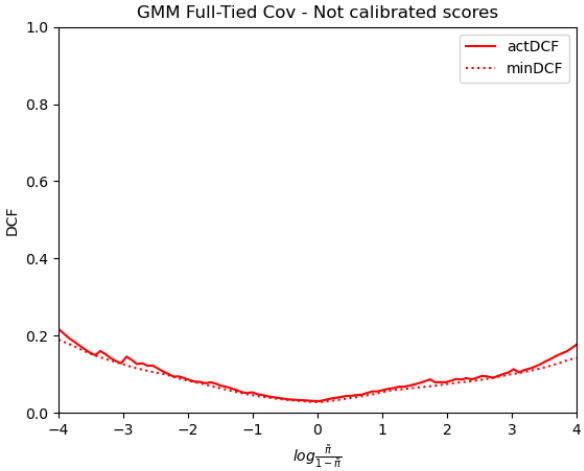
Up to now we considered only minimum detection costs as a metric to evaluate the different models, however the cost that we pay depends on a threshold we use to perform class assignment, this metric is known as actual DCF. If scores are calibrated, the optimal threshold is the theoretical one: $t = -log\frac{\tilde{\pi}}{1-\tilde{\pi}}$. But if the scores lack of a probabilistic interpretation, it is possible to compute a transformation function that maps the classifiers scores s to well-calibrated scores: $s_{cal} = f(s)$. The approach that we will employ is the one using Logistic Regression, and so that function f will have a simple form:

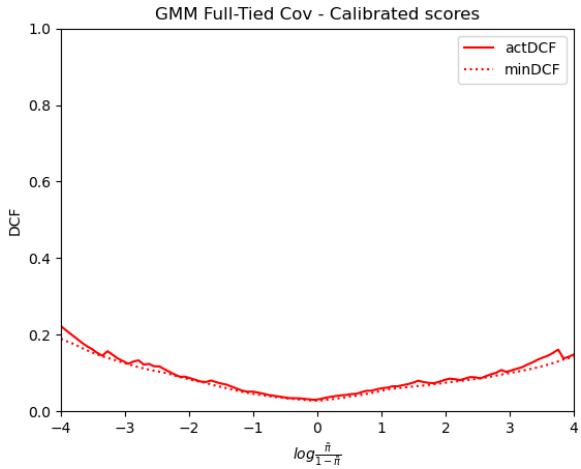
$$f(s) = \alpha s + \beta$$

Since the output of the Logistic Regression acts as posterior log-likelihood ratio we will recover the calibrated score compensating it subtracting $log\frac{\pi_t}{1-\pi_t}$. So, following this approach we need to specify a prior π_t to optimize the calibration for that specific application. Anyway, for the property of the Logistic Regression, the calibration parameters obtained using that value are able to provide good performance also for different applications (NOTE: Since the "feature" space is mono-dimensional, we don't need to use the regularized version of the Logistic Regression model, thus $\lambda = 0$ will be set).

Since we had used the K-Fold cross validation approach, and for score calibration we need to estimate only two parameters, it is reasonable to use in this case a single split approach. So, we will firstly shuffle the score, and then we will split them into two partitions: the first one will contain the 50% of the score, that will be used to estimate the parameters of the calibration function, and the remaining one will be calibrated in order to assess if the transformation is actually effective.

The GMM Full-Tied Covariance model on Z-Scored data is the best promising one, so we will calibrate its scores.





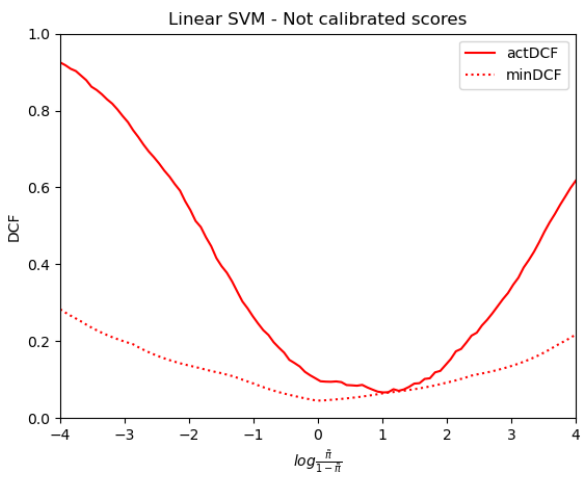
The scores do not need calibration; indeed, the transformation seems to bring any particular benefit to the model. As we can see from the results below only the unbalanced application with $\tilde{\pi} = 0.9$ slightly improved.

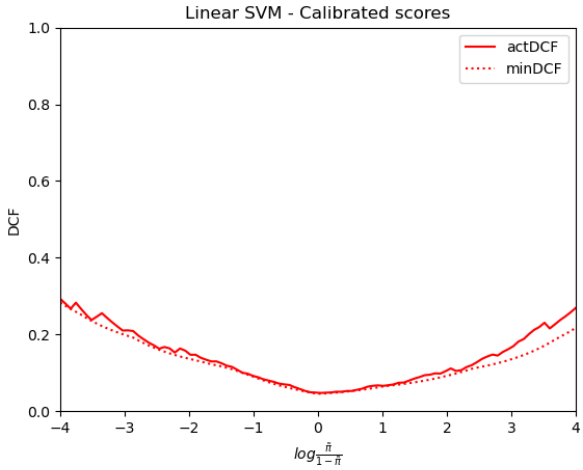
minDCF	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
-	0.027	0.092	0.078

actDCF	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Uncalibrated	0.029	0.092	0.090
Log-Reg ($\pi_t = 0.5$)	0.030	0.093	0.080

Since the performance between the uncalibrated scores and the calibrated ones do not differ too much, for simplicity we'll kept the uncalibrated model as a final classifier to deliver.

Just for educational purposes, we will consider the calibration of SVM (Z-Score data) which was one of the best promising models.





The SVM scores are not calibrated at all, so this phase would be extremely useful if we had chosen this model as a candidate. Indeed:

minDCF	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
-	0.045	0.145	0.098

actDCF	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Uncalibrated	0.098	0.605	0.182
Log-Reg ($\pi_t = 0.5$)	0.048	0.156	0.106

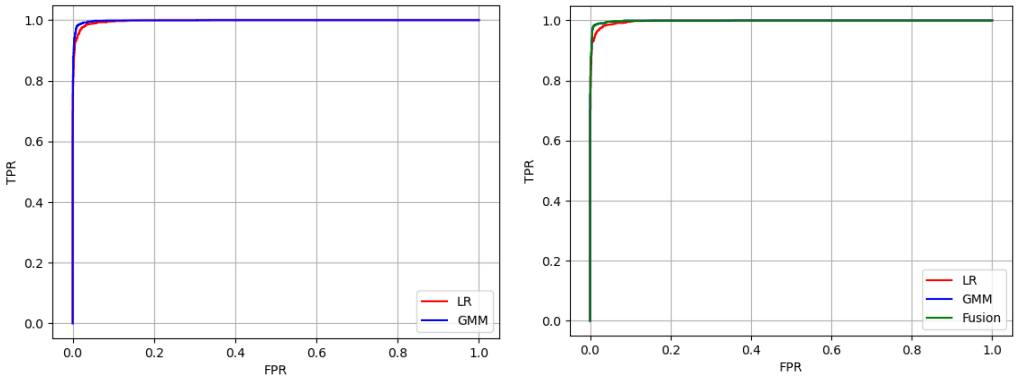
3.1 Fusion

The method used for calibration allows easily combining the outputs of different classifiers. Different can agree on some, but not all decisions, especially when classifiers use different approaches. Combining the decisions of two classifiers in this case may lead to improvement. In our case since the Full Tied GMM model outperforms for all the three considered applications every classifier, we will combine it with the linear Logistic Regression model only for educational purposes. We have chosen this model as a candidate for the fusion, since not considering the Full GMM model (which use a very similar approach of the Full Tied one), can be considered the second-best model analysed.

We assume that the fused score is a linear function of the scores of the different classifiers:

$$s_t = f(s_{t,A}, s_{t,B})$$

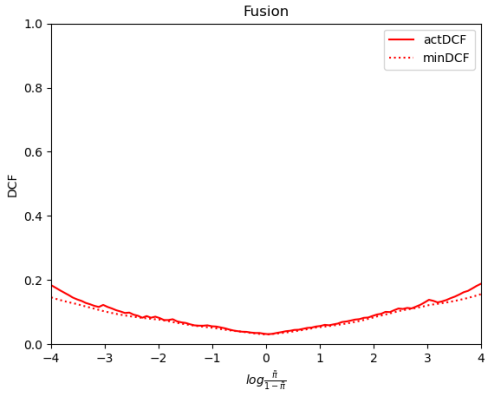
Where s_t is the fused score for sample x_t, while $s_{t,A}$ and $s_{t,B}$ are the scores of that sample generated by the classifier A (GMM) and B (LR). Sinche the function that we are searching is linear we can employ again the prior-weighted logistic regression to learn the parameter. Since the feature space is small, the values of the parameters to be estimated are only 3; so, for the training of this meta-model will be used the single split validation methodology (50% - 50%). (NOTE: for the same reason of the score calibration λ will be set to 0).
The results will be assessed with ROC plots; on the left the ROC plot computed over the whole set of score of the two classifiers, on the right the one computed over the validation portion of the fusing approach).



As it can be observed the GMM model performs on its own better than the LR one, for every operating point, thus the fused system did not bring any benefit (the Fusion line and the GMM one overlaps), indeed:

minDCF	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Tied GMM (4 comp)	0.026	0.082	0.083
Log-Reg ($\pi_t = 0.5,\ \lambda = 0$)	0.049	0.110	0.127
Fusion	0.027	0.078	0.082

Despite there is a small improvement for the unbalanced application $\tilde{\pi} = 0.1$, there is not a particular reason to prefer the fused system over the GMM one. Most likely the weight associated to the GMM score is much higher than the one associated to the LR one, so for simplicity we will keep the GMM model alone as best candidate. (Indeed, the estimated weight vector used for mapping is about [0.88, 0.06]).
The fused scores also benefit from the re-calibration effects of the meta-model. Indeed:



Since this model performs good as the GMM model and the scores of this one is already calibrated, the above results were expected also for this reason.

4 Experimental results

We now need to assess the quality of our model on the held-out data. We will verify the performance of the choice we made, to see how they affected performance for unseen data. We again evaluate systems in term of minimum DCF.

4.1 Gaussian models

Gaussianization	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Full Covariance	0.073	0.201	0.182
Full-Tied Covariance	0.069	0.184	0.177
Diagonal Covariance	0.547	0.791	0.845
Tied-Diagonal Covariance	0.545	0.793	0.846

Gaussianization – PCA (m = 11)	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Full Covariance	0.074	0.216	0.206
Full-Tied Covariance	0.079	0.205	0.203
Diagonal Covariance	0.085	0.245	0.232
Tied-Diagonal Covariance	0.084	0.236	0.232

Gaussianization – PCA (m = 10)	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Full Covariance	0.072	0.205	0.203
Full-Tied Covariance	0.078	0.199	0.205
Diagonal Covariance	0.087	0.243	0.229
Tied-Diagonal Covariance	0.084	0.234	0.225

Gaussianization – PCA (m = 9)	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Full Covariance	0.088	0.240	0.234
Full-Tied Covariance	0.097	0.258	0.231
Diagonal Covariance	0.102	0.290	0.248
Tied-Diagonal Covariance	0.104	0.286	0.258

Z-Score	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Full Covariance	0.053	0.134	0.1375
Full-Tied Covariance	0.050	0.132	0.135
Diagonal Covariance	0.570	0.810	0.882
Tied-Diagonal Covariance	0.570	0.808	0.880

Z-Score – PCA (m = 11)	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Full Covariance	0.113	0.279	0.255
Full-Tied Covariance	0.111	0.278	0.255
Diagonal Covariance	0.116	0.293	0.262
Tied-Diagonal Covariance	0.115	0.285	0.226

Z-Score – PCA (m = 10)	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Full Covariance	0.131	0.313	0.298
Full-Tied Covariance	0.129	0.311	0.294
Diagonal Covariance	0.130	0.333	0.302
Tied-Diagonal Covariance	0.127	0.330	0.303

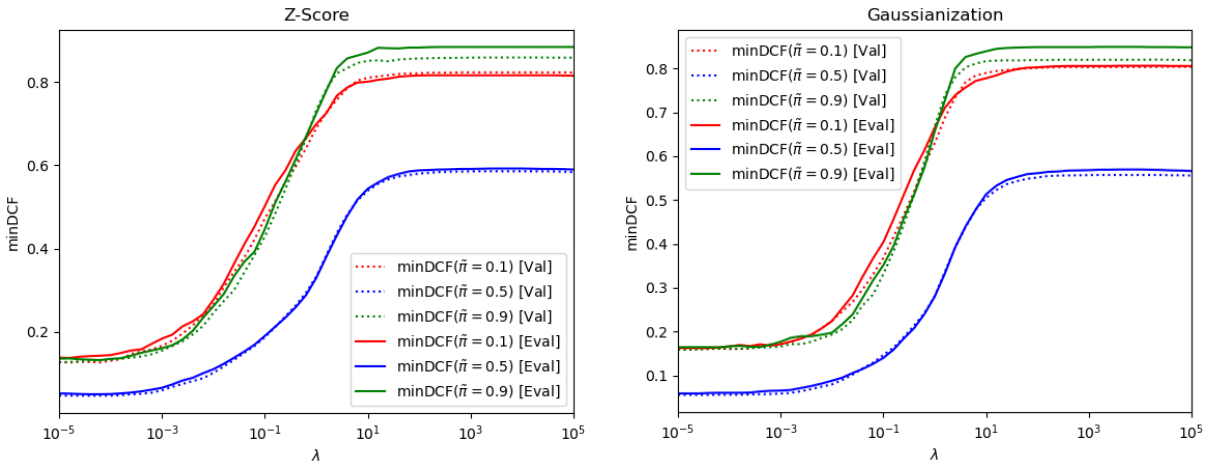
Z-Score – PCA (m = 9)	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Full Covariance	0.183	0.427	0.443
Full-Tied Covariance	0.180	0.424	0.432
Diagonal Covariance	0.184	0.447	0.420
Tied-Diagonal Covariance	0.181	0.434	0.417

The results are consistent with those obtained on the validation set. The Full-Tied Covariance model over the Z-Scored features is confirmed to be the best. PCA is not effective and the gaussianized features performs worse than the Z-Scored ones.

From the above results we can think that the evaluation population is distributed as the validation one, so the following models will behave equally as before.

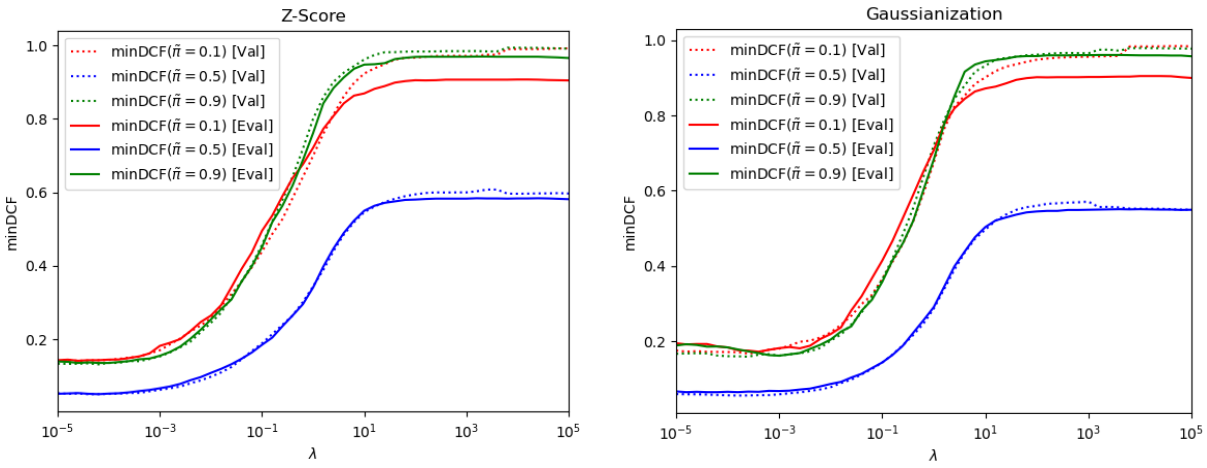
4.2 Logistic Regression models

We now consider linear logistic regression models plotting how the value λ affects the minimum DCF. (NOTE: will be considered only models trained with $\pi_t = 0.5$)



The curves for the validation and evaluation set have the same trend on overlap substantially for every value of λ, independently from the targeting application. The performances using $\lambda = 10^{-4}$ or not using regularization at all, are the same. It is confirmed that the regularization term provides no benefit. (NOTE: we will not provide any numerical results in a tabular way since the behaviour of the curves is the same, it would be substantially a "copy and paste" of the previously showed table…)

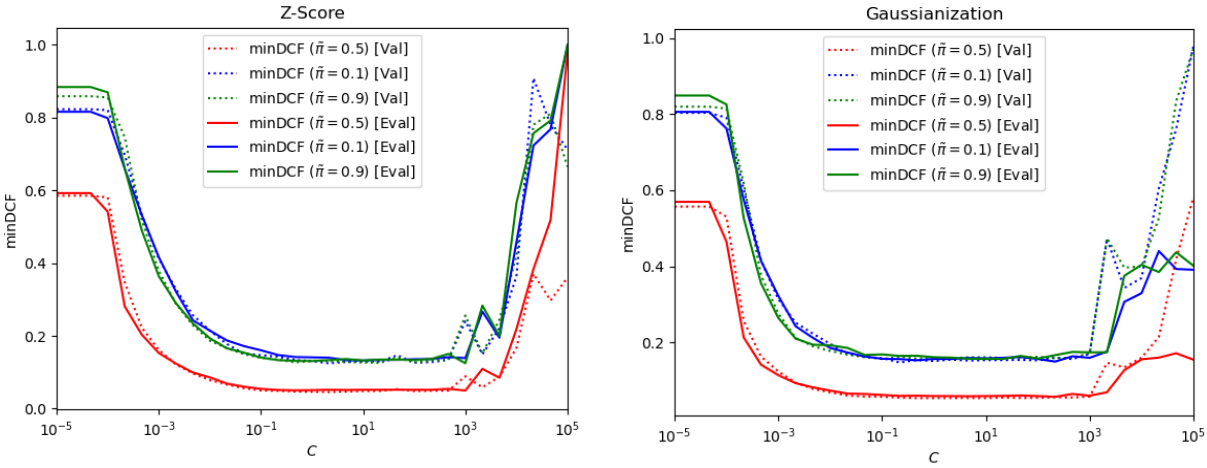
We repeat the analysis the analysis for Quadratic Logistic Regression:



Event in this case the trend of the curves is the same; so, the choice of no regularization for the Z-Score data and a value of $\lambda = 10^{-4}$ for the Gaussianized feature, was effective again.

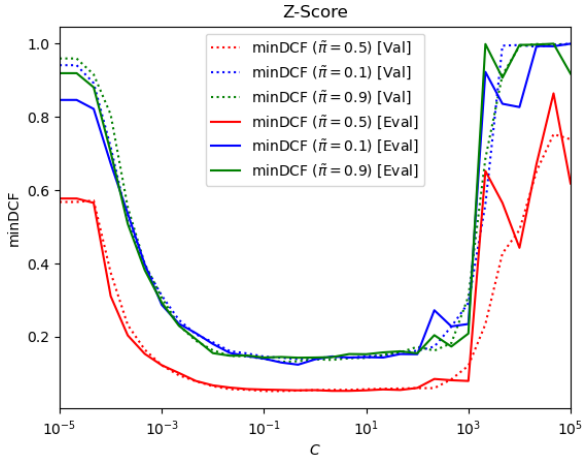
4.3 Support Vector Machine models

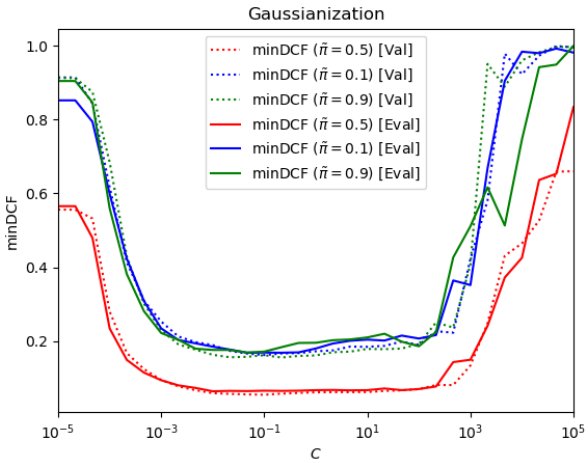
The next models are the SVM ones. We will start with the linear version. (NOTE: again, for simplicity, will be considered only models trained with $\pi_t = 0.5$)

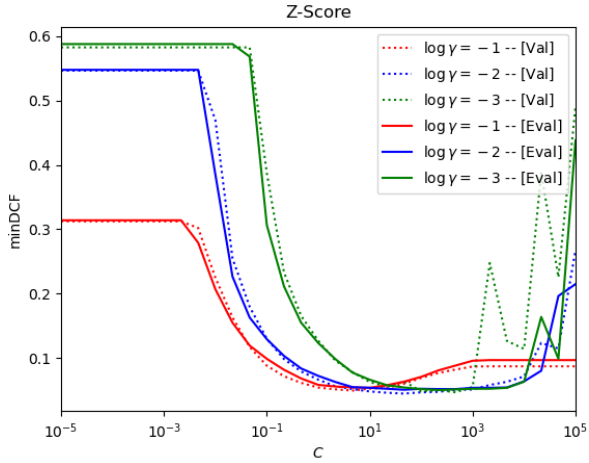


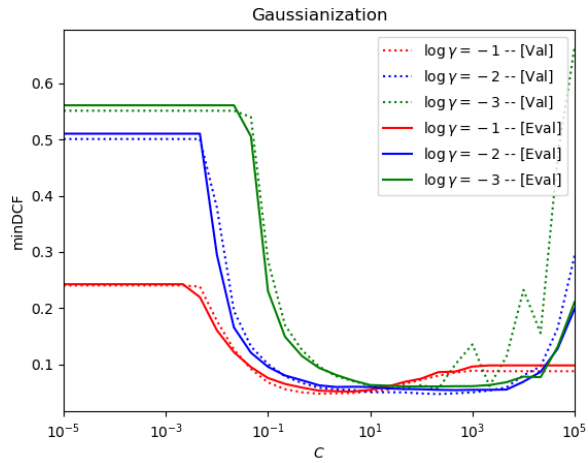
The results are in line with expectations, the choice of the hyper-parameter C was effective for both Z-Scored data and Gaussianized ones. The trend of the lines differs only for value of C greater than 10^3, however those values are not interesting.

Now we proceed with the Quadratic (top) and the Radial Basis (bottom) versions.





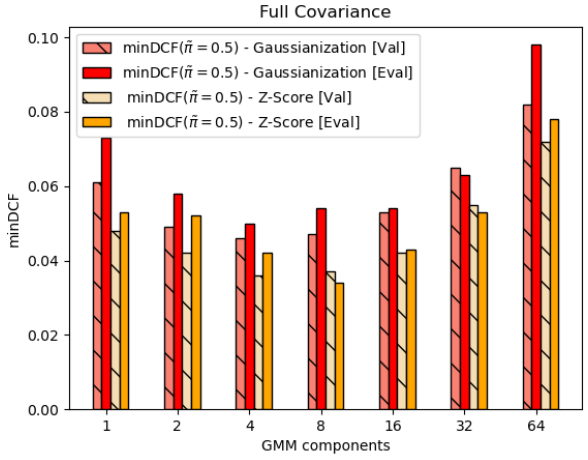


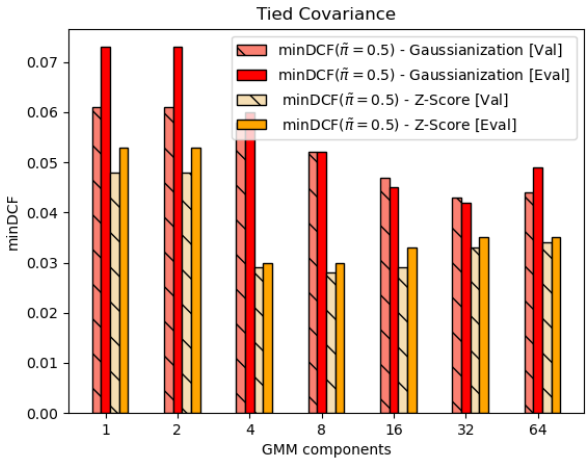


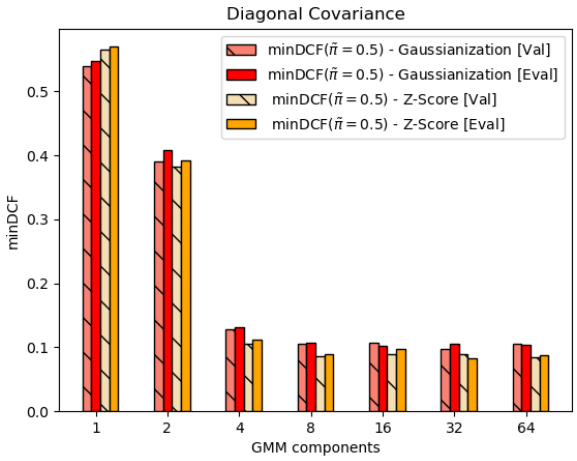
The effect of using different value of C and γ (for the Radial Basis models), are consistent with that observed on the validation data, except for value of C greater than 10^3 (especially for the Radial version).

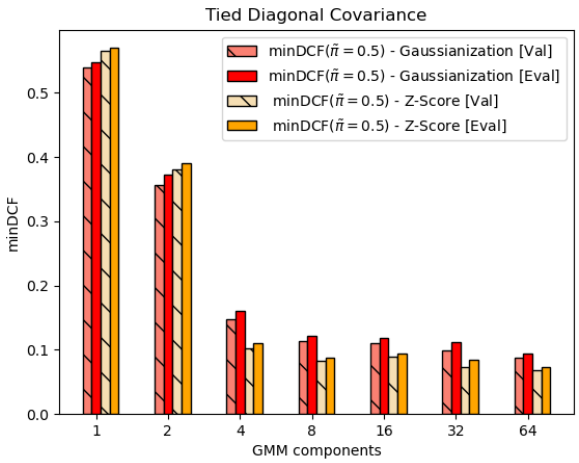
4.4 Gaussian Mixture models

The last family of classifiers that we will consider, is the GMM one.









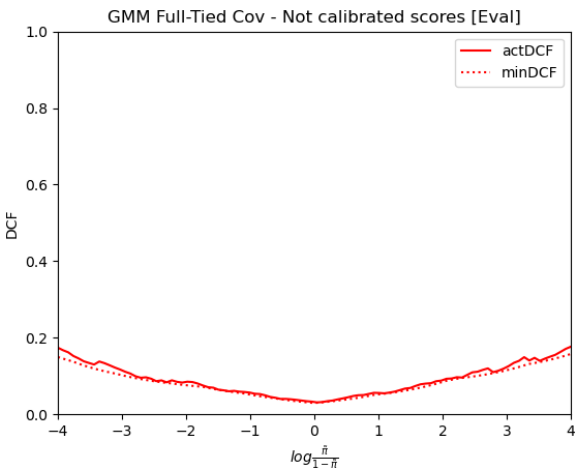
The validation results are again consistent with the evaluation ones.

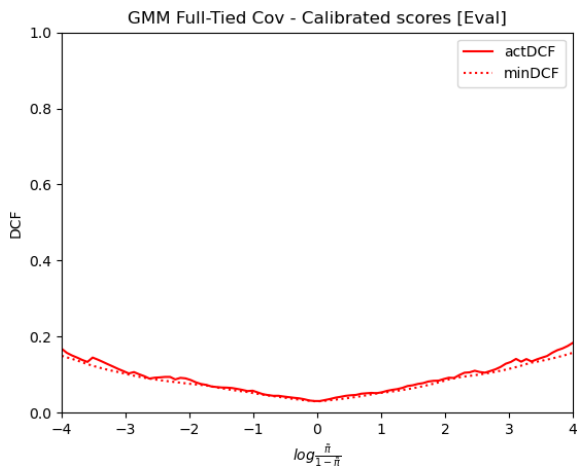
Z-Score [Eval]	$\tilde{\pi}=0.5$	$\tilde{\pi}=0.1$	$\tilde{\pi}=0.9$
Full Covariance (8 comp)	0.034	0.092	0.086
Full-Tied Covariance (4 comp)	0.029	0.080	0.092
Diagonal Covariance (32 comp)	0.083	0.244	0.248
Tied-Diagonal Covariance (64 comp)	0.073	0.180	0.204

Gaussianization [Eval]	$\tilde{\pi}=0.5$	$\tilde{\pi}=0.1$	$\tilde{\pi}=0.9$
Full Covariance (4 comp)	0.050	0.149	0.147
Full-Tied Covariance (32 comp)	0.041	0.124	0.134
Diagonal Covariance (16 comp)	0.102	0.241	0.269
Tied-Diagonal Covariance (64 comp)	0.094	0.282	0.291

For every model the number of components is equal between evaluation and validation population. For the Full Tied Covariance model, the choice between 4 and 8 number of components it turned out to be not so much important, since if we had chosen 8 the min DCF ($\tilde{\pi}=0.5$) would have been 0.030. The best model for the unbalanced application $\tilde{\pi}=0.9$ is the Full Covariance one with 4 components, which "wins" for a very few points against the Full Tied one.

The Full Tied GMM confirms to be the best promising classifier, but we need to assess if its scores need to be calibrated.





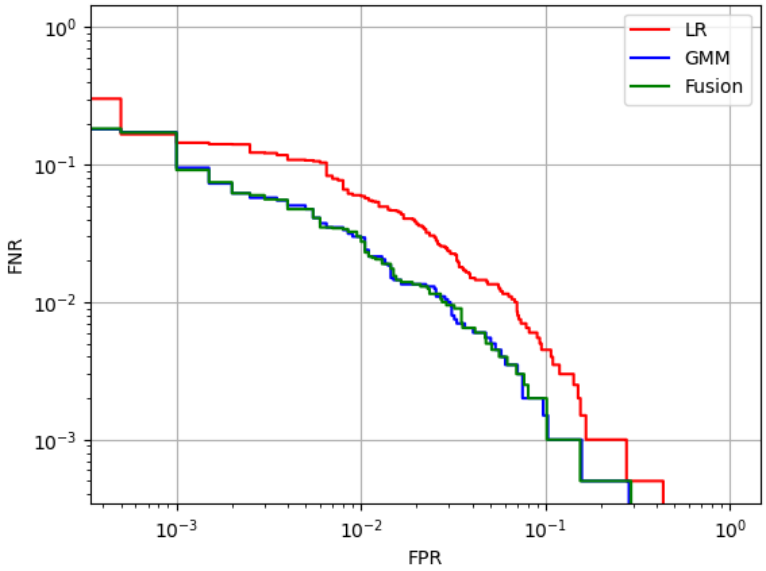
The scores have already a probabilistic interpretation, so the choice to not calibrate is not harmful. Indeed:

minDCF [Eval]	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
-	0.029	0.080	0.092

actDCF [Eval]	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Uncalibrated	0.031	0.088	0.096
Log-Reg ($\pi_t = 0.5$)	0.030	0.095	0.095

4.5 Fusion

The last analysis involves the Fusion model (Tied GMM + linear LR) on Z-Scored data. Here is the DET curve plot:



As we expected the, the fusion system aligns with the GMM one, giving the same results. Indeed:

Z-Score	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Tied GMM (4 comp)	0.029	0.080	0.092
Log-Reg ($\pi_t = 0.5,\ \lambda = 0$)	0.052	0.137	0.135
Fusion	0.030	0.080	0.091

5 Conclusions

The similarity between the validation and evaluation set strongly suggests that the evaluation population is distributed as the training one. All the choice we made on our training data proved to be effective also on the test one.

The GMM Full Tied Covariance model (on Z-Scored data) with 4 components is effective and produce well-calibrate scores substantially for every kind of application. We can achieve a very low DCF cost of 0.031 for our primal application ($\tilde{\pi} = 0.5$) and decent ones of 0.088 and 0.096 for the secondary scenarios (respectively $\tilde{\pi} = 0.1$ and $\tilde{\pi} = 0.9$).