POLITECNICO DI TORINO
MACHINE LEARNING AND PATTERN RECOGNITION
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

**Machine Learning and Pattern Recognition**

---

**Report type h**

# Gender Voice Detection

---

Author: Serra Matteo S303513

,

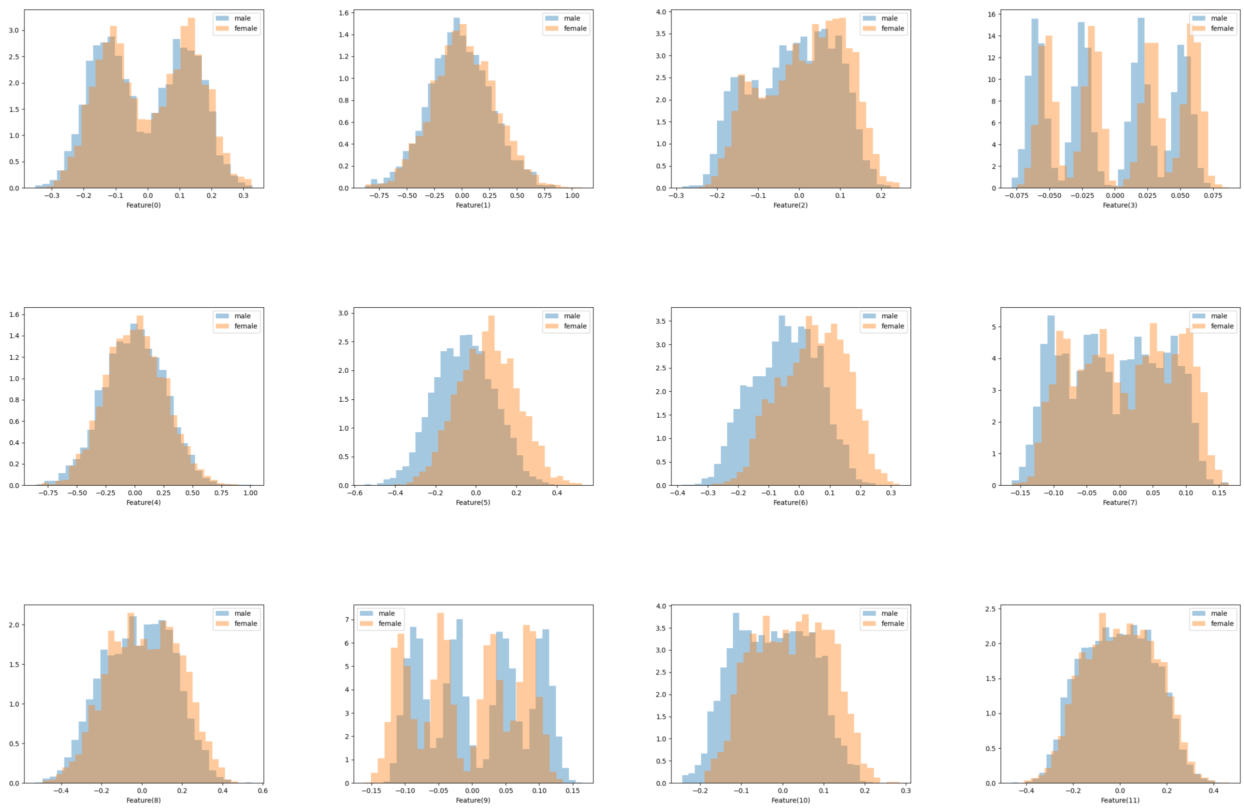
# Contents

# 1 Introduction

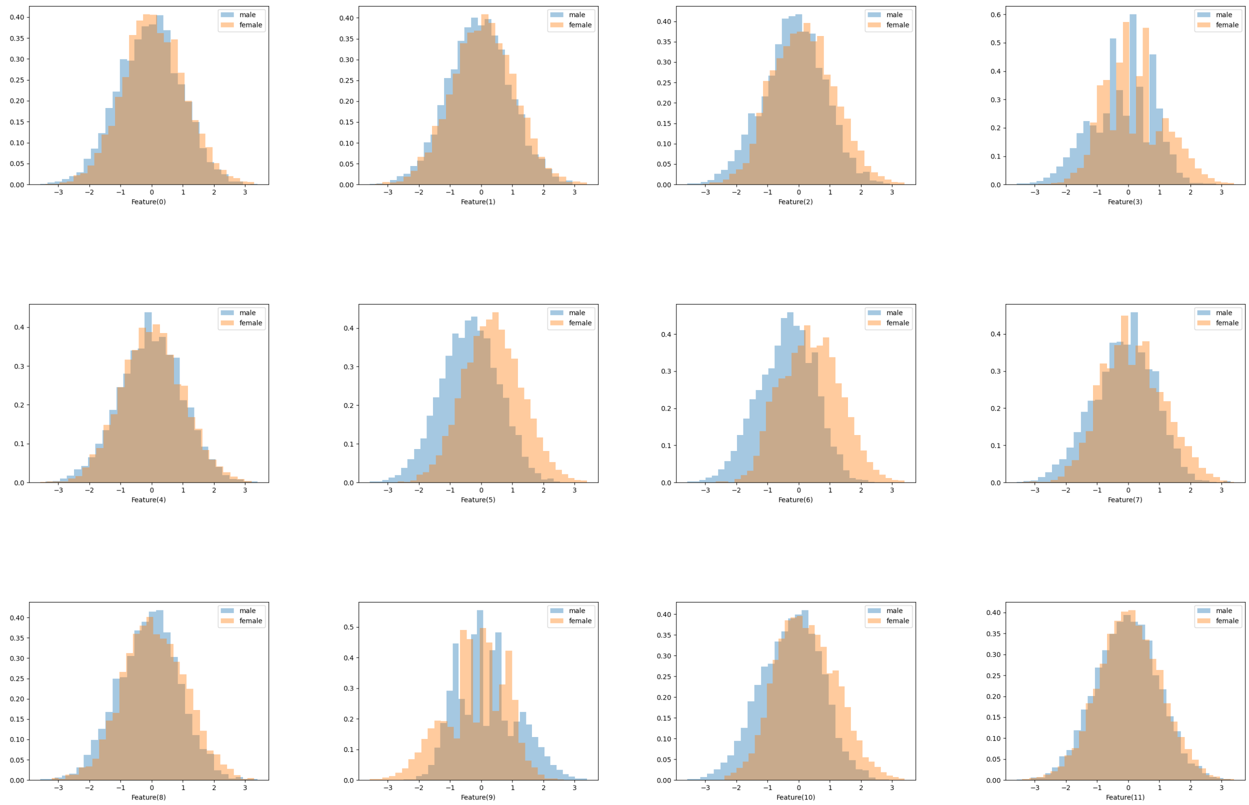
The dataset consists of speaker embeddings that represent the acoustic of a spoken utterance. Each row corresponds to a different speaker and contains 12 features followed by the gender label (1 for female, 0 for male). The features do not have any particular interpretation. Speakers belong to four different age groups. The age information, however, is not available.

The training set consists of 3000 samples per class, whereas the test set contains 2000 samples per class. To make the problem more tractable, and to avoid potential privacy issues, the dataset consists of synthetic samples that behave similarly to real speaker embeddings. Features are continuous values that represent a point in the m-dimensional embeddings space.

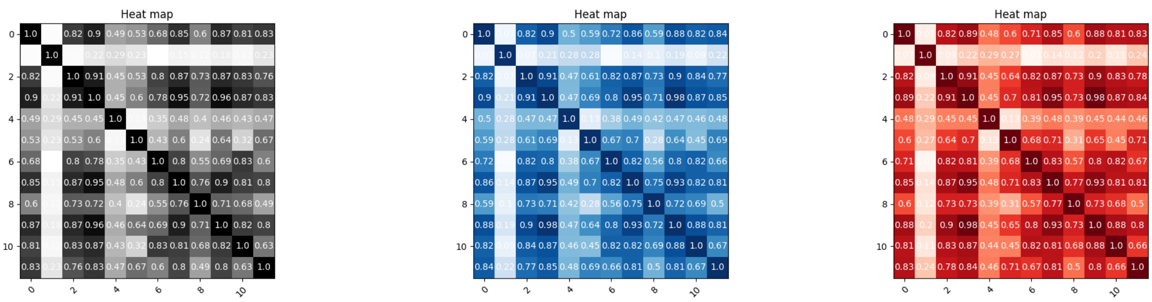
Below we can see the histograms of the dataset features (training set) after the z-normalization.




**Gaussianized features** We repeat the analysis also with the gaussianization technique on the raw features and see if there is some improvements in the performances.




As we can see from the plots above the Gaussianization transformation doesn't change the features distribution shapes except for features 3 and 9, with this preprocessing method we lose the information about the gaussian cluster distribution. A correlation analysis of z-normalized features shows that some features are strongly correlated (in this case we assume a strong correlation for values greater than 0.85). Heat maps show the absolute value of the Pearson correlation coefficient: gray: whole dataset, blue: samples of male voices, red: samples of female voices.




This suggests that we may benefit from using PCA to map data to 8 uncorrelated features to reduce the number of parameters to estimate.

## 2 Gaussian Classifiers

We start considering gaussian classifiers, since within-class covariance matrices are far from being diagonal we will expect not optimal results from the diagonal covariance approach. To understand which model is more promising, and to assess the effectiveness of using PCA we adopt the K-fold cross-validation method with K = 4, in this way the single fold will consist of 75% of the development data and 25% of validation data. Data has been shuffled before splitting, so that the data of different folds are homogeneous. Our application will be a uniform prior one:

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

But we will also consider unbalanced applications:

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

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

Where the prior is biased towards one of the two classes. In this preliminary analysis we will focus on the min DCF metric, after that we will compute the optimal threshold for each method. The selection of the m parameter for the PCA is related to the previous observations done on the heat map plots.

| | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.5$ |
|---|---|---|---|
| Z-normalized features - no PCA | | | |
| Full-Cov | 0.0476 | 0.1246 | 0.1273 |
| Diag-Cov | 0.5593 | 0.8303 | 0.8633 |
| Tied Full-Cov | 0.0467 | 0.1233 | 0.1246 |
| Tied-Diag-Cov | 0.5637 | 0.8280 | 0.8553 |
| Z-normalized features - PCA (m=9) | | | |
| Full-Cov | 0.0470 | 0.1327 | 0.1213 |
| Diag-Cov | 0.0683 | 0.1700 | 0.1597 |
| Tied Full-Cov | 0.0473 | 0.1277 | 0.1206 |
| Tied Diag-Cov | 0.0673 | 0.1600 | 0.1597 |
| Z-normalized features - PCA (m=8) | | | |
| Full-Cov | 0.0477 | 0.1370 | 0.1297 |
| Diag-Cov | 0.0686 | 0.1680 | 0.1673 |
| Tied Full-Cov | 0.0483 | 0.1277 | 0.1290 |
| Tied Diag-Cov | 0.0676 | 0.1627 | 0.1670 |

| | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.5$ |
|---|---|---|---|
| Raw features - no PCA | | | |
| Full-Cov | 0.0487 | 0.1246 | 0.1273 |
| Diag-Cov | 0.5593 | 0.8303 | 0.8633 |
| Tied Full-Cov | 0.0467 | 0.1233 | 0.1246 |
| Tied Diag-Cov | 0.5637 | 0.8280 | 0.8533 |
| Gaussianization - no PCA | | | |
| Full-Cov | 0.0630 | 0.1900 | 0.1747 |
| Diag-Cov | 0.5387 | 0.8173 | 0.8337 |
| Tied Full-Cov | 0.0613 | 0.1783 | 0.1700 |
| Tied Diag-Cov | 0.5347 | 0.8010 | 0.8173 |
| Gaussianization - PCA=9 | | | |
| Full-Cov | 0.092 | 0.2483 | 0.2410 |
| Diag-Cov | 0.0973 | 0.2647 | 0.2573 |
| Tied Full-Cov | 0.0897 | 0.2410 | 0.2373 |
| Tied Diag-Cov | 0.098 | 0.2570 | 0.2603 |

**Conclusions** Overall, the MVG model with tied covariance performs better, we have to say that the bad results that come from the diagonal covariance models are caused by the strong correlation between features, the diagonal matrices are far from being diagonal.

From the table above we can see that the PCA is not effective either for Full or Diagonal covariance models, on the contrary it makes the models results even worse. The full covariance models perform in general slightly worse.

Gaussianization doesn't improve the classification, so from now we choose to use only the z normalization as preprocessing step.

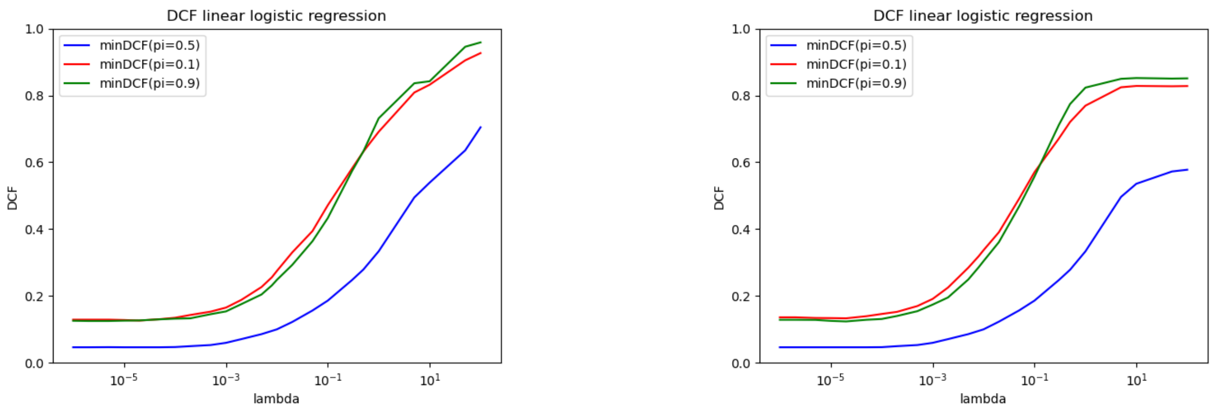
Overall, the best candidate is currently the MVG model with tied Covariance matrice without PCA and as we can see from the table above the results are also good for imbalanced applications. Given the limited effectiveness of PCA for generative models we only consider using the whole set of features. We can also notice that the use of PCA makes the effectivness of the Diagonal models better, we think that this is due to the fact that removing the more correlated features makes the covariance matrix more diagonal.

# 3 Logistic Regression

Our classes are balanced, so re-balancing the cost of the different classes it's not strictly mandatory. However, we try to re-balanced the costs of the two classes, minimizing:

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

We start considering a prior of $\pi_T = 0.5$ and compute the minDCF for each value of lambda in order to tune the hyper parameter $\lambda$. To do that we use a K-fold approach over a validation set, the best value corresponds to the one that has the lowest value of minDCF.

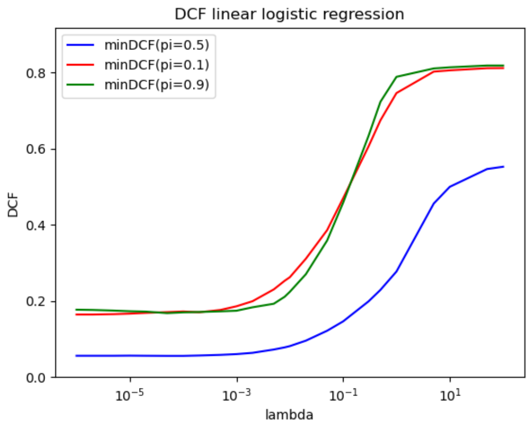



(a) not weighted (b) weighted

As we can see from the plot the optimal value of lambda is obtained with low value, we select $\lambda = 10^{-6}$ and we can observe that for low value of lambda re-balancing the costs of the two classes is useless. We can also consider different prior $\pi_T$ to see the effect on the other applications.

| | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ |
|---|---|---|---|
| MVG | | | |
| MVG (Full-Cov) | 0.0476 | 0.1246 | 0.1273 |
| MVG (Tied Full-Cov) | 0.0467 | 0.1233 | 0.1247 |
| Z-norm Features | | | |
| Log Reg ($\lambda = 10^{-6}, \pi_T = 0.5$) | 0.0463 | 0.1297 | 0.1257 |
| Log Reg ($\lambda = 10^{-6}, \pi_T = 0.1$) | 0.0476 | 0.1353 | 0.1300 |
| Log Reg ($\lambda = 10^{-6}, \pi_T = 0.9$) | 0.1257 | 0.1277 | 0.1293 |
| Raw Features | | | |
| Log Reg ($\lambda = 10^{-6}, \pi_T = 0.5$) | 0.0463 | 0.1297 | 0.1257 |
| Log Reg ($\lambda = 10^{-6}, \pi_T = 0.1$) | 0.0476 | 0.2967 | 0.1300 |
| Log Reg ($\lambda = 10^{-6}, \pi_T = 0.9$) | 0.0466 | 0.1277 | 0.1293 |

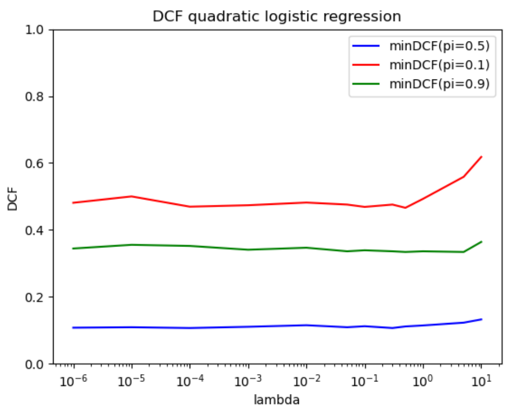
**Gaussianization** We will repeat the analysis applyng the gaussianization on raw features.



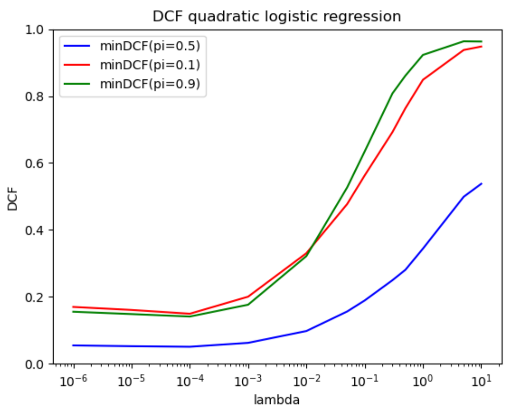

| | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ |
|---|---|---|---|
| Gauss Features | | | |
| Log Reg ($\lambda = 10^{-6}, \pi_T = 0.5$) | 0.0563 | 0.1643 | 0.1647 |
| Log Reg ($\lambda = 10^{-6}, \pi_T = 0.1$) | 0.0590 | 0.1667 | 0.1637 |
| Log Reg ($\lambda = 10^{-6}, \pi_T = 0.9$) | 0.0560 | 0.1680 | 0.1730 |

Overall, the logistic regression linear model perform slightly better for our main application, while as before the gaussianization does not improve the classification. We repeat the analysis for Quadratic Logistic Regression.

## 3.1 Quadratic Logistic Regression




(a) quadratic logreg with raw features (balanced)

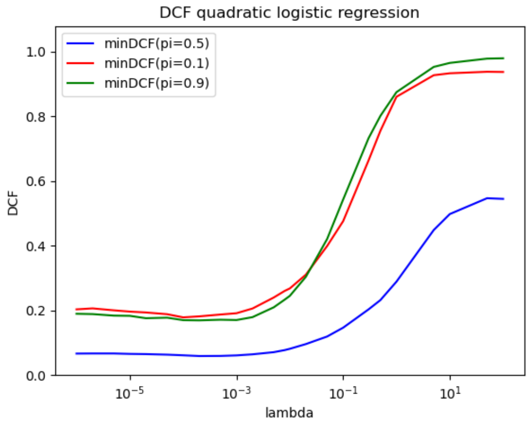



(b) quadratic logreg with z-norm features (balanced)

Again, we consider training using different prior to see the effects on the other applications.

| | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ |
|---|---|---|---|
| Raw Features | | | |
| MVG (Tied Full-Cov) | 0.0467 | 0.1233 | 0.1246 |
| Quad Log Reg ($\lambda = 10^{-6}, \pi_T = 0.5$) | 0.0463 | 0.1297 | 0.1267 |
| Quad Log Reg ($\lambda = 10^{-6}, \pi_T = 0.1$) | 0.0476 | 0.1353 | 0.1276 |
| Quad Log Reg ($\lambda = 10^{-6}, \pi_T = 0.9$) | 0.0467 | 0.1300 | 0.1294 |
| Z-norm features | | | |
| MVG (Tied Full-Cov) | 0.0467 | 0.1233 | 0.1246 |
| Quad Log Reg ($\lambda = 10^{-6}, \pi_T = 0.5$) | 0.0546 | 0.1450 | 0.1423 |
| Quad Log Reg ($\lambda = 10^{-6}, \pi_T = 0.1$) | 0.0577 | 0.1697 | 0.1510 |
| Quad Log Reg ($\lambda = 10^{-6}, \pi_T = 0.9$) | 0.0569 | 0.1556 | 0.1533 |

**Gaussianization** We repeat the experiment with the gaussianization preprocessing enabled.



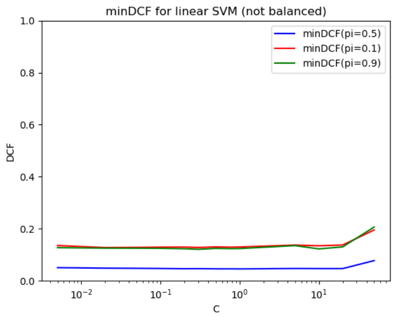

| | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ |
|---|---|---|---|
| Raw Features | | | |
| Quad Log Reg ($\lambda = 10^{-6}, \pi_T = 0.5$) | 0.0657 | 0.1733 | 0.1600 |
| Quad Log Reg ($\lambda = 10^{-6}, \pi_T = 0.1$) | 0.0658 | 0.1963 | 0.1790 |
| Quad Log Reg ($\lambda = 10^{-6}, \pi_T = 0.9$) | 0.0670 | 0.1659 | 0.1833 |

**Conclusions** As before, a good value for lambda is $10^{-6}$ but as we can see from the table above, the quadratic logistic regression does not improve the performances of the log reg model. Compared to the MVG models the min DCF becomes better for those applications with unbalanced class distribution. Overall we can observe that the linear models seems to perform better then the others. We will now focus on SVM and GMM classifiers. As before we assist a degradation of the performances with the application of the gaussianization preprocessing.

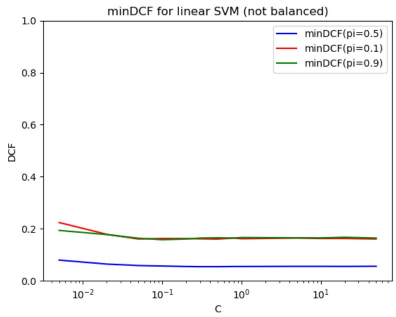
# 4 SVM

## 4.1 Linear SVM

For linear SVM, we need to tune the hyper-parameter C. Again, we use the k-fold cross validation method to find the near-optimal value of C. We start with a model that does not balance the two classes.




(a) linear SVM




(b) linear SVM with Gauss Features

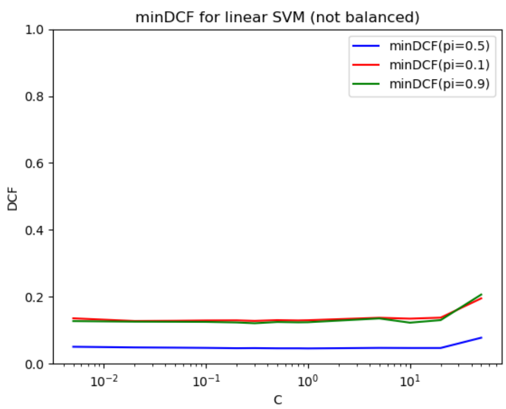
To rebalance the classes, we use a different value of C for the different classes

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

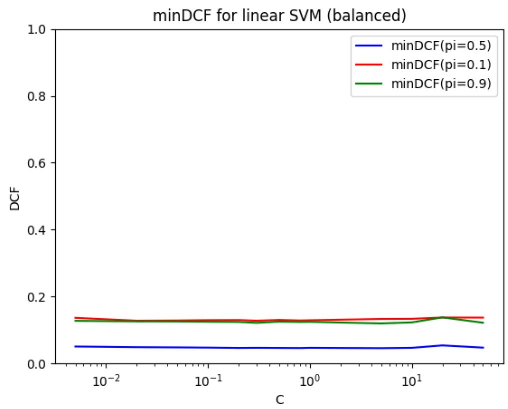
subject to

$$0 \leq \alpha_i \leq C_i, i = \ ..., n$$

Where the i-th C corresponds to $C_T$ for samples of the class $H_T$ or to $C_F$ for samples of the other class. Since we are not modelling the bias term, we omitted the constant related to bias. So, we select $C_T = C\frac{\pi_T}{\pi_T^{emp}}$ and $C_F = C\frac{\pi_T}{\pi_T^{emp}}$ where $\pi_F^{emp}$ and $\pi_T^{emp}$ are the empirical priors for the two classes computed over the training set.




(a) linear SVM without class rebalancing



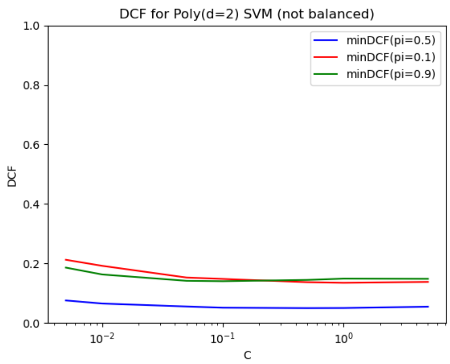

(b) linear SVM with class rebalancing

As we can see from the graphics above the class rebalancing doesn't improve the performance of the linear model. The choice of C does not look critical, we select C = 1 because from that point the value of the duality gap starts to increase, and the approximation becomes less accurate. We can compare linear models in terms of min DCF:

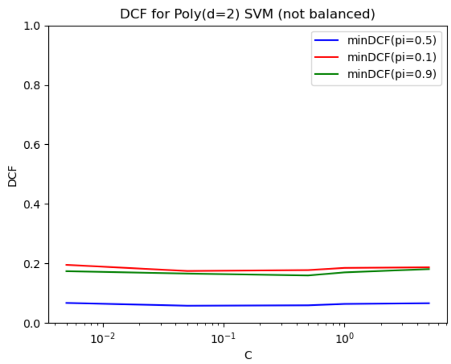
| | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ |
|---|---|---|---|
| Raw features | | | |
| MVG (Tied Full-Cov) | 0.0467 | 0.1233 | 0.1246 |
| Log Reg ($\lambda = 10^{-6}, \pi_T = 0.5$) | 0.0463 | 0.1297 | 0.1257 |
| Linear SVM (C=1) | 0.0500 | 0.1300 | 0.1287 |
| Linear SVM (C=1, $\pi_T = 0.5$) | 0.0460 | 0.1287 | 0.1317 |
| Z-norm features | | | |
| Linear SVM (C=1) | 0.0500 | 0.1300 | 0.1287 |
| Linear SVM (C=1, $\pi_T = 0.5$) | 0.0477 | 0.1303 | 0.1270 |
| Gauss features | | | |
| Linear SVM (C=1) | 0.0550 | 0.1617 | 0.1663 |
| Linear SVM (C=1, $\pi_T = 0.5$) | 0.0550 | 0.1613 | 0.1663 |

## 4.2 Poly SVM

We are going to consider two non-linear SVM models. The first will use a polynomial quadratic kernel, for this one we expect similar results respect the Quadratic Logistic Regression models. The second will employ a radial basis function kernel. For the quadratic kernel we have to estimate the value of C, in order to do this, we use the k-fold approach and evaluate the min DCF for each value of C.




(a) Poly SVM - znorm features



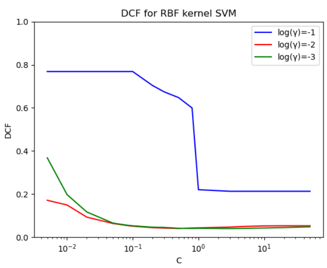

(b) Poly SVM with Gauss Features

Choice of C does not look critical for this reason we chose C = 0.1. We can now compare quadratic models in terms of minDCF

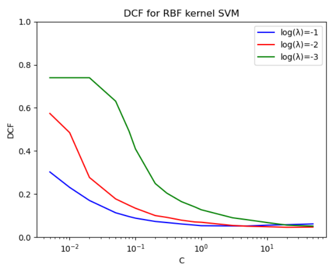
| | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ |
|---|---|---|---|
| MVG (Full-Cov) | 0.0476 | 0.1246 | 0.1273 |
| Quad Log Reg ($\lambda = 10^{-6}, \pi_T = 0.5$) | 0.0533 | 0.1463 | 0.1437 |
| Z-norm features | | | |
| Quadratic SVM (C=1) | 0.0507 | 0.1487 | 0.1407 |
| Gauss features | | | |
| Quadratic SVM (C=1) | 0.0640 | 0.1840 | 0.1703 |

**Conclusions** Comparing different quadratic classifiers we see that the best performances are reached with the MVG with full covariance model.

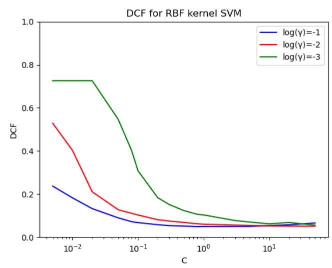
## 4.3 RBF SVM




(a) RBF SVM - Raw features




(b) RBF SVM - Znorm Features




(c) RBF SVM - Gauss Features

The plot shows that both $\gamma$ and C influence the results. For raw feautures best results are obtained using log $\gamma$ = 0.001 and C=1, while for the preprocessed models we can observe that an optimal choice is taking $\gamma$ = 0.1 and C=1. Class re-balancing for the primary task (using the same values for C and $\gamma$ as for imbalanced model) provides very similar result since the dataset is already class balanced.

| | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ |
|---|---|---|---|
| MVG (Full-Cov) | 0.0476 | 0.1246 | 0.1273 |
| Quad Log Reg ($\lambda = 10^{-6}, \pi_T = 0.5$) | 0.0533 | 0.1463 | 0.1437 |
| Quadratic SVM (C=0.1) | 0.0513 | 0.1423 | 0.159 |
| Raw features | | | |
| RBF SVM ($C = 1, \log\gamma = -3$) | 0.0413 | 0.1253 | 0.1187 |
| Z-norm features | | | |
| RBF SVM ($C = 1, \log\gamma = -1$) | 0.0413 | 0.1253 | 0.1187 |
| Gauss features | | | |
| RBF SVM ($C = 1, \log\gamma = -1$) | 0.0493 | 0.1453 | 0.1253 |

We try with class rebalancing in order to see if it helps the performance on different applications.

| | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ |
|---|---|---|---|
| Raw features | | | |
| RBF SVM ($C = 1, \log \gamma = -3$) | 0.0413 | 0.1253 | 0.1187 |
| RBF SVM ($C = 1, \log \gamma = -3, \pi_T = 0.5$) | 0.0413 | 0.1253 | 0.1187 |
| RBF SVM ($C = 1, \log \gamma = -3, \pi_T = 0.5$) | 0.0419 | 0.1319 | 0.1337 |
| RBF SVM ($C = 1, \log \gamma = -3, \pi_T = 0.5$) | 0.0430 | 0.1357 | 0.1277 |

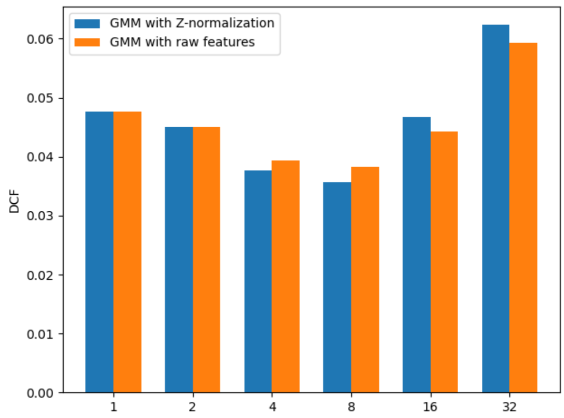
As we can see, the chosen gamma is still the better and the choice of C does not look critical. Overall the results are again consistent. We also repeated the analysis using the gaussianized features.

| | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ |
|---|---|---|---|
| Gauss features | | | |
| RBF SVM ($C = 1, \log \gamma = -1$) | 0.0493 | 0.1453 | 0.1253 |
| RBF SVM ($C = 1, \log \gamma = -1, \pi_T = 0.5$) | 0.0493 | 0.1453 | 0.1253 |
| RBF SVM ($C = 1, \log \gamma = -1, \pi_T = 0.1$) | 0.0556 | 0.1540 | 0.1653 |
| RBF SVM ($C = 1, \log \gamma = -1, \pi_T = 0.9$) | 0.0553 | 0.1653 | 0.1440 |

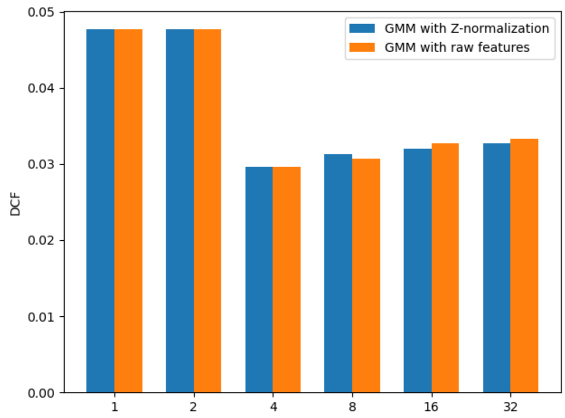
**Conclusions** As we can see from the results above the linear SVM performs pretty well and we can assume at this point that the linear models performs better than the quadratic ones. Overall the RBF model is the best model since now.

## 5 Gaussian Mixture Models

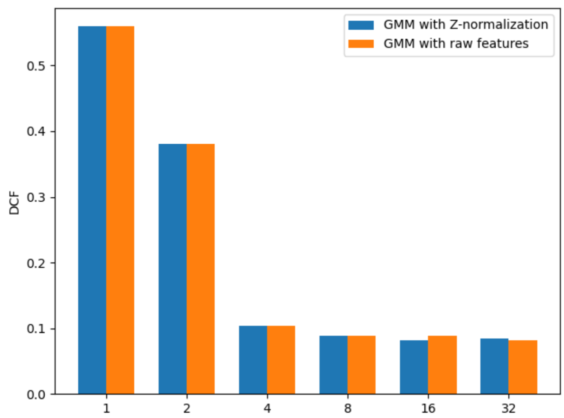
The last model we are going to consider is a generative approach based on training a GMM over the data of each class. Since GMMs can approximate generic distributions, we expect to obtain better results than with the Gaussian model. In this case the hyperparameter to be tuned is the number of component C. In this section we consider all the combination of model, so we examine both full covariance and diagonal models, with and without covariance tying. Also in this case we use the K-fold protocol to select the number of Gaussians and to compare different models.



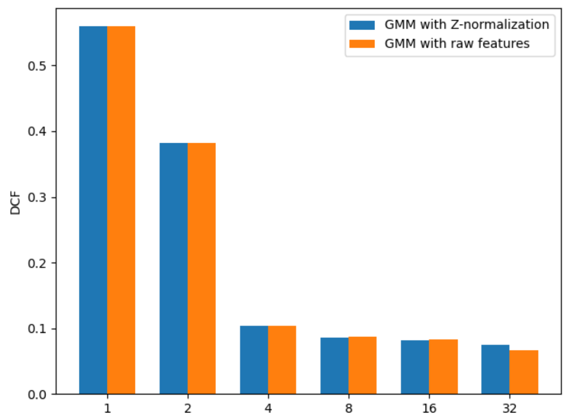

(a) Full covariance - Non Tied Model




(b) Full covariance - Tied Model




(a) Diagonal covariance - Non Tied Model




(b) Diagonal covariance - Tied Model

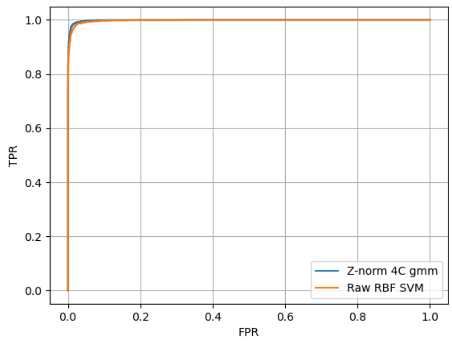
As we can see from the plots and more in details from the table below, the best full covariance models have similar performance (both best min DCFs are around 0.03 - 0.04 ) while we can see a drop in performance with diagonal models. Furthermore the results with Z-normalization compared with the raw features ones are pretty similar.

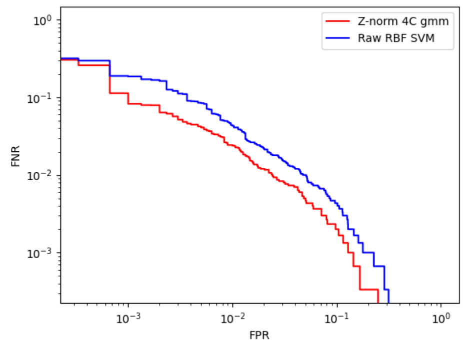
| | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ |
|---|---|---|---|
| Raw features | | | |
| RBF SVM (C=1, $\log\gamma = -3, \pi_T = 0.5$) | 0.0413 | 0.1253 | 0.1187 |
| RBF SVM (C=1, $\log\gamma = -3, \pi_T = 0.1$) | 0.0419 | 0.1319 | 0.1337 |
| RBF SVM (C=1, $\log\gamma = -3, \pi_T = 0.9$) | 0.0430 | 0.1357 | 0.1277 |
| GMM Full-Cov, 8 Gau | 0.0337 | 0.0846 | 0.0830 |
| GMM Diag-Cov, 8 Gau | 0.3522 | 0.0826 | 0.0803 |
| GMM Tied Full-Cov, 4 Gau | 0.0296 | 0.0855 | 0.0832 |
| GMM Tied-Diag-Cov, 32 Gau | 0.3550 | 0.0820 | 0.0806 |
| Z-normalized features | | | |
| GMM Full-Cov, 8 Gau | 0.0383 | 0.0904 | 0.0909 |
| GMM Tied Full-Cov, 4 Gau | 0.0296 | 0.0879 | 0.0878 |

**Conclusions** The poor performance of the Diagonal Covariance models can be explained by the high correlation between some features as we can see from the heat map in the introduction section. From the table above we can see that this method outperform the RBF SVM and it is the best in terms of minDCF.

# 6 Model Evaluation and Calibration

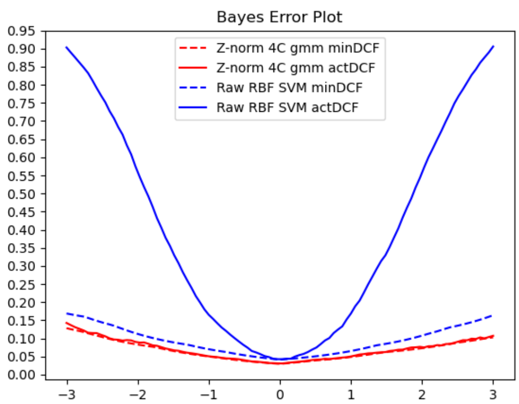
We can evaluate the performance of our classifiers plotting the DET plot or a ROC plot in order to see how well do they perform in different operating points. In particular we are going to compare our 2 best models since now: the RBF SVM with $\gamma = 0.001$ and the Tied Covariance GMM with 4 components.




So we select as candidate model the GMM with tied full covariance matrix with 4 components. As secondary we will take in consideration the RBF SVM with $\gamma = 0.001$. Up to now we have considered only minimum DCF metrics. Min DCF measures the cost we would pay if we made optimal decisions for the evaluation set using the recognizer scores. The cost that we actually pay, however, depends on the goodness of the decisions we make using those scores (in the binary case, on the goodness of the threshold we use in practice to perform class assignment). We therefore turn our attention to actual DCFs

| | $\tilde{\pi} = 0.5$ | | $\tilde{\pi} = 0.1$ | | $\tilde{\pi} = 0.9$ | |
|---|---|---|---|---|---|---|
| | min DCF | act DCF | min DCF | act DCF | min DCF | act DCF |
| GMM tied 4C | 0.02967 | 0.0307 | 0.0900 | 0.0957 | 0.0783 | 0.0807 |
| RBF SVM($\gamma = 0.001$) | 0.0413 | 0.0423 | 0.1253 | 0.6567 | 0.1187 | 0.6423 |



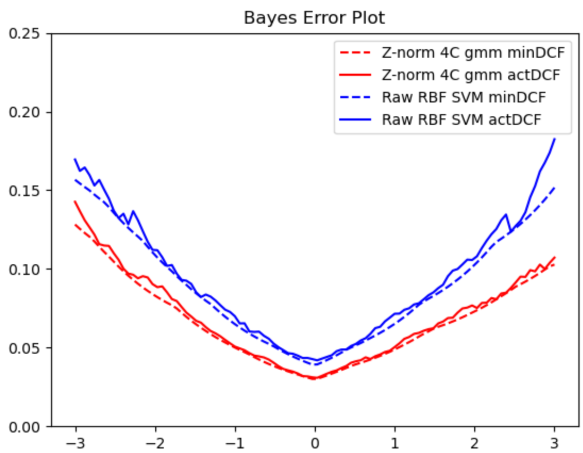
As we can see from the Bayes Error Plot the RBF SVM scores are highly unbalanced, while the GMM scores are almost calibrated. The explanation to this is that the SVM model lack a probabilistic interpretation. Our solution consists in re-calibrating the scores in a manner that the theoretical threshold $t = -\log \frac{\tilde{\pi}}{1-\tilde{\pi}}$ provides close to optimal values over a wide range of effective priors $\tilde{\pi}$. To do this we employed the prior-weighted Logistic Regression model to learn the model parameters over our training scores. To recover the calibrated score $f(s)$ we will need to compute:

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

The results with the re-calibrated scores can be seen below:

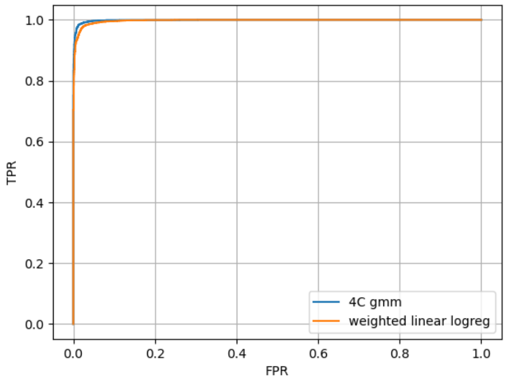
| | min DCF($\tilde{\pi} = 0.5$) | min DCF($\tilde{\pi} = 0.1$) | min DCF($\tilde{\pi} = 0.9$) |
|---|---|---|---|
| | 0.0414 | 0.1278 | 0.1260 |
| | act DCF($\tilde{\pi} = 0.5$) | act DCF($\tilde{\pi} = 0.1$) | act DCF($\tilde{\pi} = 0.9$) |
| Uncalibrated | 0.0440 | 0.1769 | 0.1730 |
| LogReg($\lambda = 0$) | 0.0420 | 0.1319 | 0.1278 |

The recalibration process improves the performance of our classifier bringing the maximum difference between minDCF and actDCF to 3%. We can see the effect of the re-calibration also in the Bayes Error Plot below

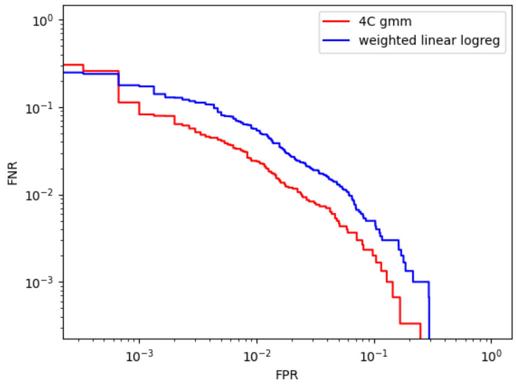



**GMM 4C Tied Covariance vs Linear Logistic Regression** In this section we show the comparison in terms of minDCF and actDCF between the best logistic regression model, the linear one, and the overall best model: the 4C GMM with full tied covariance.

| | $\tilde{\pi} = 0.5$ | | $\tilde{\pi} = 0.1$ | | $\tilde{\pi} = 0.9$ | |
|---|---|---|---|---|---|---|
| | min DCF | act DCF | min DCF | act DCF | min DCF | act DCF |
| GMM tied 4C | 0.02967 | 0.0307 | 0.0900 | 0.0957 | 0.0783 | 0.0807 |
| Linear LogReg | 0.0463 | 0.0473 | 0.1297 | 0.1327 | 0.1257 | 0.1313 |

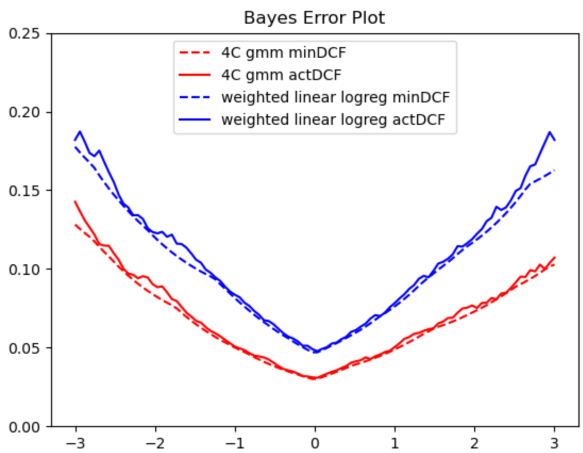



(a) ROC plot




(b) DET plot

Especially the DET plot confirms that the GMM model is superior to logistic regression for many operating points except for some at the beginning.

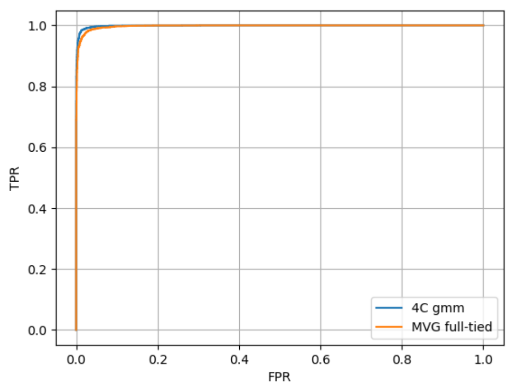



As we can see from the plot above, the scores of the weighted linear logistic regression and the 4 components GMM are already calibrated so they don't need a score calibration. We can now consider scores of other methods and see how they behave.

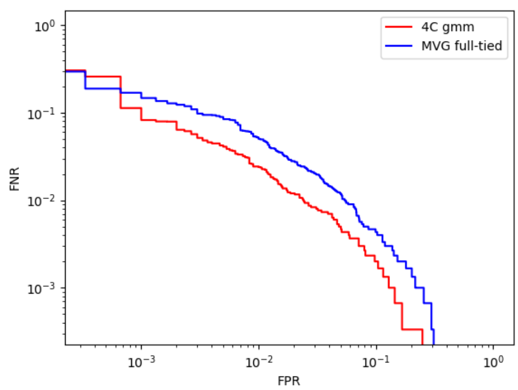
**GMM 4C Tied Covariance vs MVG Full Tied Covariance** The same is true if we compare log likelihood ratios of the same GMM model that we have seen before and the MVG with full tied covariance.

| | $\tilde{\pi} = 0.5$ | | $\tilde{\pi} = 0.1$ | | $\tilde{\pi} = 0.9$ | |
|---|---|---|---|---|---|---|
| | min DCF | act DCF | min DCF | act DCF | min DCF | act DCF |
| GMM tied 4C | 0.0296 | 0.0307 | 0.0900 | 0.0957 | 0.0783 | 0.0807 |
| MVG Full Tied | 0.0467 | 0.0487 | 0.1233 | 0.1253 | 0.1247 | 0.1330 |

As we can see from the plots below the GMM with 4 components with tied covariance is superior for many operating points respect to the correspondent MVG classifier. But taking in count that the MVG is the method that is better in terms of DET plot for more operating points respect the other methods, we will use it in a fused approach in order to improve both methods.

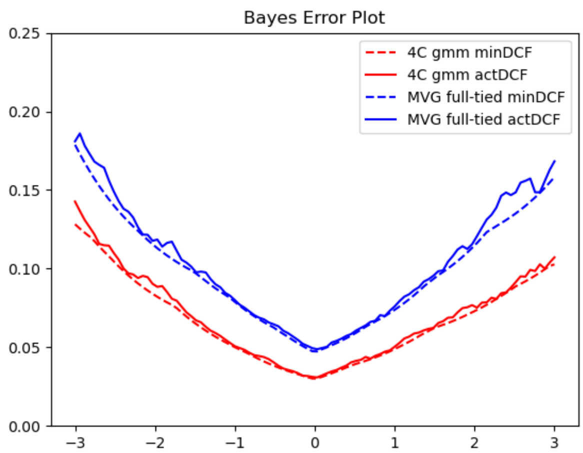



(a) ROC plot




(b) DET plot

The Error Bayes plot and the table show that the scores are well calibrated and don't need a recalibration like the SVM method.



# 7 Model Fusion

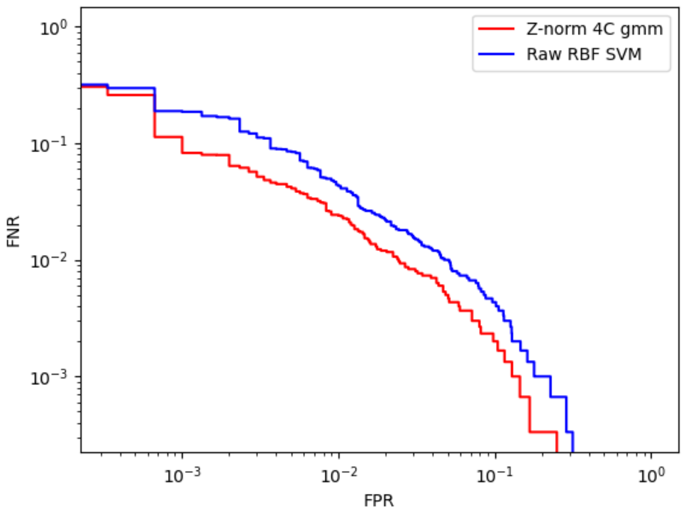
In order to improve the global performances we decide to use the model fusion method: it allows easily combining the outputs of different classifiers. Combining the decisions of two classifiers in this case may lead to improvement. Our decision leads to perform score-level fusion, rather than decision-level voting. We assume that the fused score is a function of the scores of the different classifiers. If $s_{t,A}$ and $s_{t,B}$ are the scores of classifiers A and B for sample $x_t$, then the fused score for sample $x_t$ will be:

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

We will then use the fused scores to perform the decision. As for calibration we can assume a simple linear form for $f$:

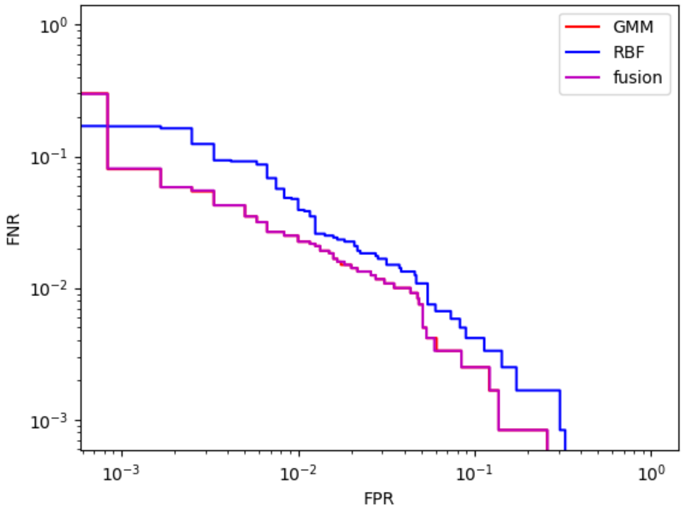
$$f(s_{t,A}, s_{t,B}) = \alpha_A s_{t,A} + \alpha_B s_{t,B} + \beta$$

Again, we can use prior-weighted logistic regression to train the model parameters. Looking at the the previous results we thought of using the RBF model as second model with the GMM 4C Full-Tied.

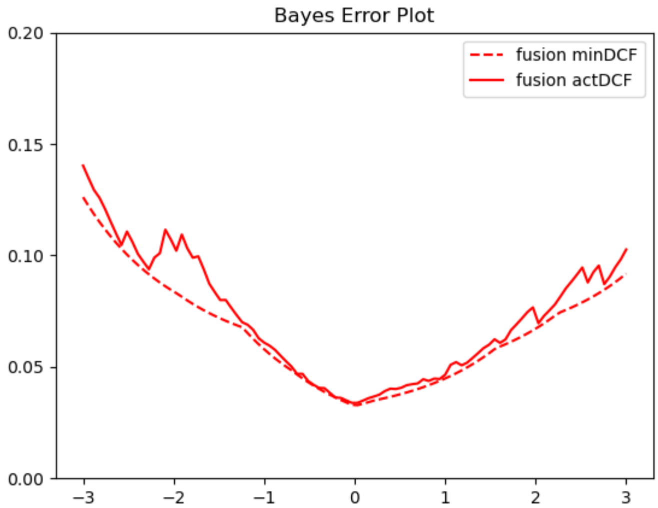



(a) 4C gmm - RBF SVM

| | $\tilde{\pi}_T = 0.5$ | | $\tilde{\pi}_T = 0.9$ | | $\tilde{\pi}_T = 0.1$ | |
|---|---|---|---|---|---|---|
| | minDCF | actDCF | minDCF | actDCF | minDCF | actDCF |
| RBF SVM | 0.0383 | 0.04083 | 0.1200 | 0.652 | 0.1233 | 0.6458 |
| GMM 4C Tied Full | 0.0325 | 0.0333 | 0.0908 | 0.1033 | 0.0725 | 0.07666 |
| Fusion | 0.0325 | 0.0333 | 0.08917 | 0.0975 | 0.0725 | 0.0775 |




(a) 4C gmm - RBF SVM - fusion




(b) fusion - Bayes Error Plot

**Conclusions** The results shows that the fusion system follows the GMM 4C Tied Full Covariance we think this is due to the fact that the linear regression model assign a higher weight to the GMM model and nearly ingnores the SVM model. We can see that this method improves the unbalanced applications, but not the main one.

# 8 Experimental Results

We start again from the Gaussian Classifiers, and we again evaluate systems in terms of minimum DCFs. The minimum DCF provides an optimistic estimate of the actual DCF: it's the cost we would have if we were able to select the optimal threshold for the evaluation set. This allows veryfing whether the proposed solution is indeed the one that can achieve the best accuracy. We will then asses how good our actual decisions are for the model evaluating the actual DCF considering bot the case where we estimated and optimal threshold for the application on the validation set, and the case where scores were calibrated and the theoretical threshold was then used for different applications).

## 8.1 Multivariate Gaussian Classifier

Below are the results for the Gaussian classifiers using the training partition for training and the test partition to validate the results.

| | Z-norm features | | | Gauss features | | |
|---|---|---|---|---|---|---|
| | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ |
| No PCA | | | | | | |
| Full-Cov | 0.0530 | 0.1345 | 0.1375 | 0.0730 | 0.2015 | 0.1820 |
| Diag-Cov | 0.5700 | 0.8010 | 0.8820 | 0.5470 | 0.7915 | 0.8455 |
| Tied Full-Cov | 0.0505 | 0.1325 | 0.1350 | 0.0690 | 0.1845 | 0.1770 |
| Tied-Diag-Cov | 0.5700 | 0.8085 | 0.8795 | 0.5450 | 0.7930 | 0.8465 |
| PCA (m = 9) | | | | | | |
| Full-Cov | 0.1835 | 0.4275 | 0.4435 | 0.0985 | 0.2550 | 0.2295 |
| Diag-Cov | 0.1840 | 0.4470 | 0.4200 | 0.1020 | 0.2900 | 0.2480 |
| Tied Full-Cov | 0.1805 | 0.4240 | 0.4325 | 0.0975 | 0.2585 | 0.2315 |
| Tied-Diag-Cov | 0.1815 | 0.4345 | 0.4170 | 0.1040 | 0.2860 | 0.2580 |
| PCA (m = 8) | | | | | | |
| Full-Cov | 0.1970 | 0.4895 | 0.4510 | 0.1875 | 0.4615 | 0.4455 |
| Diag-Cov | 0.1965 | 0.5055 | 0.4340 | 0.1865 | 0.4775 | 0.4390 |
| Tied Full-Cov | 0.1950 | 0.4790 | 0.4480 | 0.1855 | 0.4655 | 0.4410 |
| Tied-Diag-Cov | 0.1965 | 0.4985 | 0.4435 | 0.1915 | 0.4755 | 0.4460 |

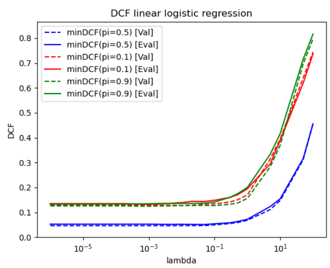
**Conclusions** The results are slightly different from those obtained with the k-fold protocol, but the best model is still the Tied Full Covariance classifier and as before the Gaussianization does not bring any improvement. The raw features results are omitted since in this classifier are exactly the same as the ones obtained with z-normalization. Also, the PCA method improves the diagonal classifiers as before.

## 8.2 Logistic Regression

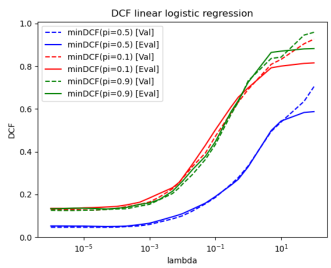
**Linear Model** We now consider linear logistic regression models with the estimated $\lambda = 10^{-6}$ value.

| | Raw features | | | Z-norm features | | | Gauss features | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ |
| LR($\pi_T = 0.5$) | 0.0525 | 0.1350 | 0.1330 | 0.0525 | 0.1350 | 0.1330 | 0.0590 | 0.1615 | 0.1645 |
| LR($\pi_T = 0.1$) | 0.0530 | 0.1425 | 0.1315 | 0.0530 | 0.1425 | 0.1315 | 0.0615 | 0.1664 | 0.1670 |
| LR($\pi_T = 0.9$) | 0.0520 | 0.1385 | 0.1360 | 0.0520 | 0.1385 | 0.1365 | 0.0620 | 0.1650 | 0.1645 |

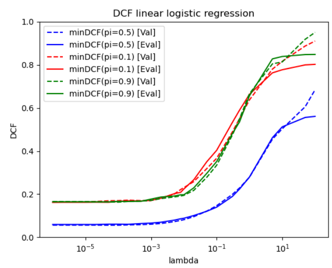
Results are consistent with our expectations. The model has similar performance of his correspondent in the training phase. Although we do not expect significant deviations, we can also verify whether the chosen $\lambda$ provides close to optimal results.




(a) Raw Features




(b) Z-Norm Features



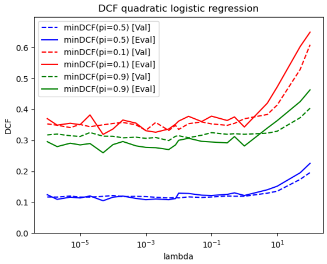

(c) Gauss Features

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 $\lambda = 10^{-6}$ has given quite remarkable results.

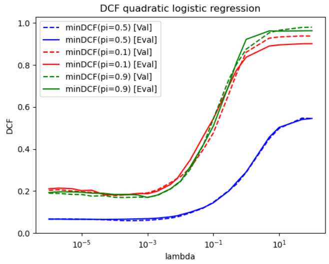
**Quadratic Model** We repeat the analysis for quadratic logistic regression, with the estimated value $\lambda = 10^{-6}$.

| | Raw features | | | Z-norm features | | | Gauss features | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ |
| QLR($\pi_T = 0.5$) | 0.1125 | 0.3525 | 0.2815 | 0.0525 | 0.1435 | 0.1380 | 0.0680 | 0.1975 | 0.1915 |
| QLR($\pi_T = 0.1$) | 0.1680 | 0.5165 | 0.3275 | 0.0570 | 0.1665 | 0.1395 | 0.0705 | 0.2165 | 0.1989 |
| QLR($\pi_T = 0.9$) | 0.1590 | 0.4295 | 0.3595 | 0.0545 | 0.1515 | 0.1500 | 0.0670 | 0.1770 | 0.1955 |

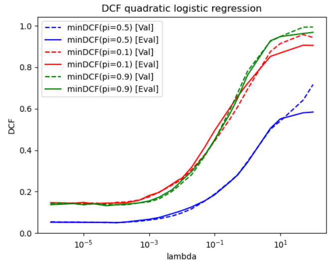
As before, results on the evaluation set are consistent with those on the validation set. Again, we can verify that our strategy for the estimation of $\lambda$ was effective:




(a) Raw features




(b) Gauss features

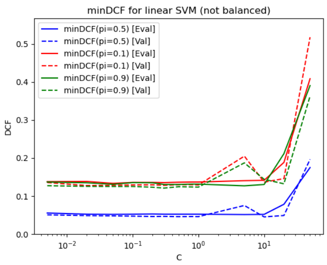



(c) Z-Norm features

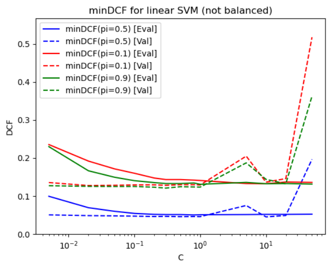
## 8.3 SVM

We repeat this analysis for each of the SVM models.

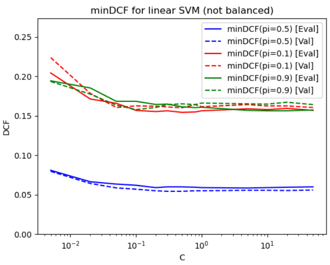
**Linear Model** For linear SVM we chose C=1, so we report for completeness also these results for raw, gaussianized and z-norm features.




(a) Raw Features



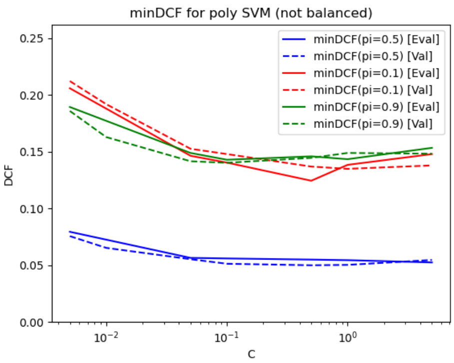

(b) Z-Norm Features



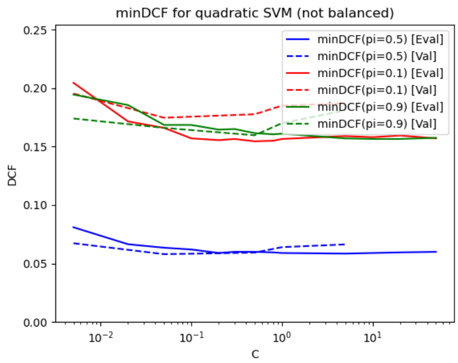

(c) Gauss Features

| | Raw features | | | Z-norm features | | | Gauss features | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ |
| Linear | 0.0525 | 0.1370 | 0.1315 | 0.0510 | 0.1410 | 0.1310 | 0.0590 | 0.1565 | 0.1610 |

**Poly (d = 2) Model** We also repeat the analysis for quadratic SVM.



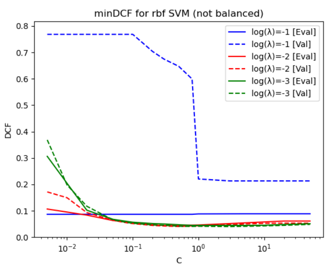

(a) Z-Norm Features



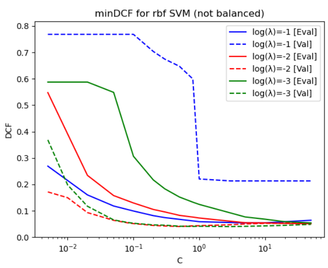

(b) Gauss Features

| | Raw features | | | Z-norm features | | | Gauss features | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ |
| Poly | 0.0620 | 0.1735 | 0.1675 | 0.0560 | 0.1405 | 0.1430 | 0.0680 | 0.1975 | 0.1915 |

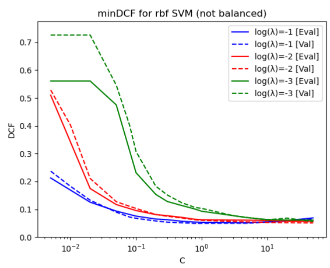
**RBF Model** We also repeat the analysis for RBF SVM.




(a) Raw Features




(b) Z-Norm Features




(c) Gauss Features

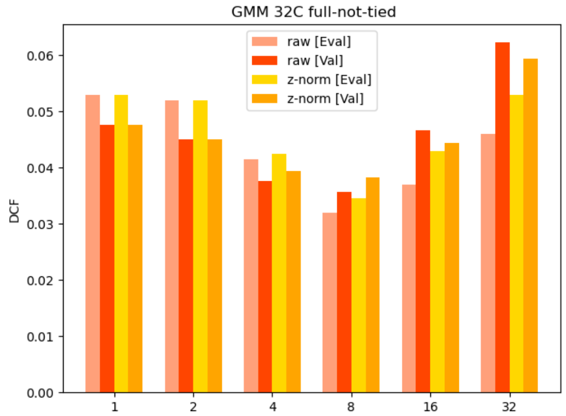
| | Raw features | | | Z-norm features | | | Gauss features | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ |
| RBF | 0.0445 | 0.1330 | 0.1120 | 0.0447 | 0.1213 | 0.1227 | 0.0530 | 0.1505 | 0.1420 |

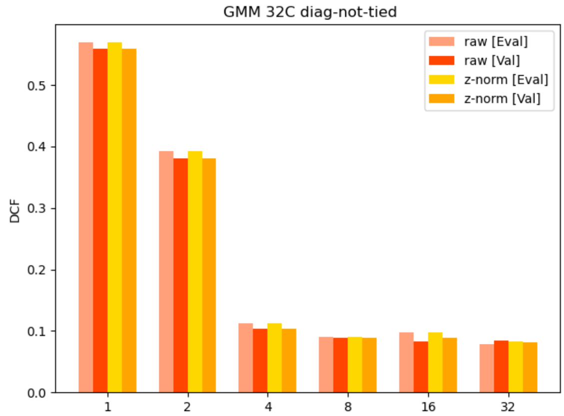
**Conclusion** As we can see both linear and quadratic model follow the same trend that they adopted during the training phase. The RBF model changed its trends during the evaluation phase but as we can see from the plots above our choices are still valid: $\gamma = 0.001$ is still the best option for the raw features classifier and the $\gamma = 0.1$ is the best one for the model that uses gaussianized features.

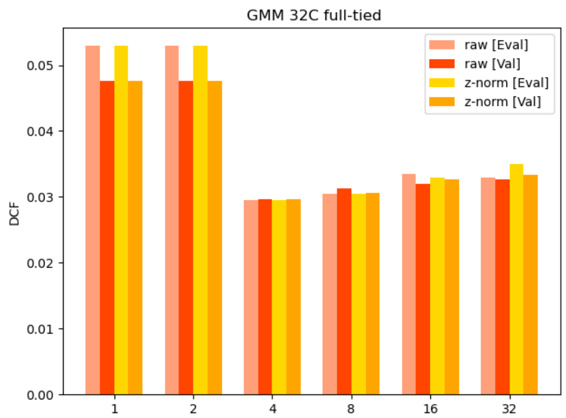
## 8.4 GMM

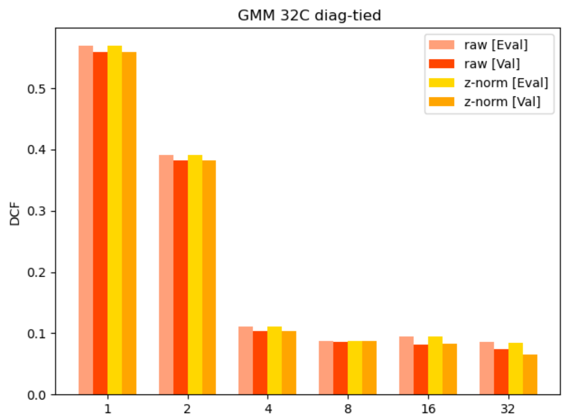
We now consider the GMM classifiers. Again we consider min DCF on the evaluation set of models trained with different number of Gaussians. Below we can see the table with the minDCF results of our classifiers.

| | Raw features | | | Z-norm features | | |
|---|---|---|---|---|---|---|
| | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.5$ |
| Full-Cov, 4C | 0.0415 | 0.114 | 0.1449 | 0.0425 | 0.1150 | 0.1170 |
| Diag-Cov, 8C | 0.1120 | 0.1258 | 0.2900 | 0.1120 | 0.2585 | 0.2900 |
| Tied Full-Cov, 4C | 0.0295 | 0.0800 | 0.0920 | 0.0295 | 0.0800 | 0.0920 |
| Tied Diag-Cov, 8C | 0.1110 | 0.2565 | 0.2945 | 0.1110 | 0.2565 | 0.2945 |

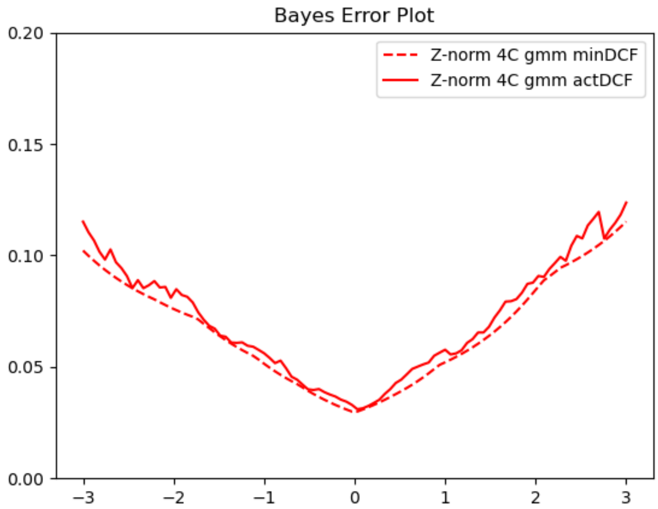



**Conclusions** As we can see from the table and the plots above the GMM model follows the training trend of the minDCF metrics, the best model is still the Full Tied Covariance model with 4 components. We can now proceed to analyze the actual DCF metric in order to access how good are the decisions that we are able to make using the recognizer score. To do this we reuse the score calibration trained with the target application $\tilde{\pi}_T = 0.5$.



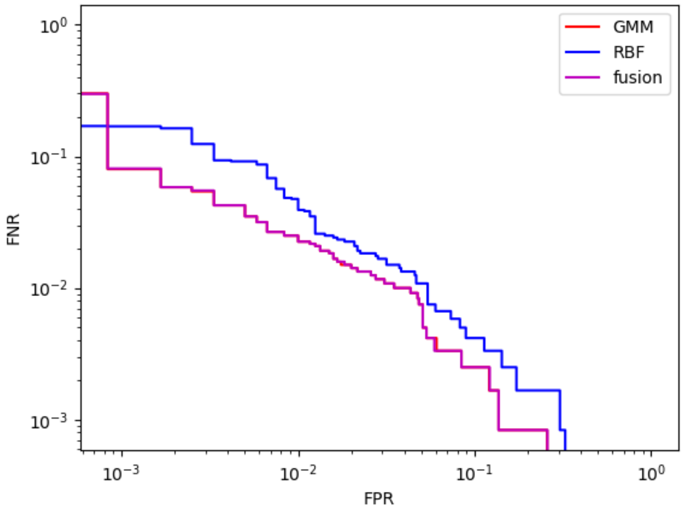

**Calibration** As we can see from the plot above the scores are already well calibrated, this is proved both from the plot and the table below.

| | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ |
|---|---|---|---|---|---|---|
| | actDCF | minDCF | actDCF | minDCF | actDCF | minDCF |
| GMM 4C Tied-Full | 0.0315 | 0.0295 | 0.088 | 0.0800 | 0.0960 | 0.0920 |

## 8.5 Fusion

Finally, we can evaluate the effectiveness of our model fusion. It still involves the Tied GMM with the RBF SVM trained on row features.

| | $\tilde{\pi} = 0.5$ | $\tilde{\pi} = 0.1$ | $\tilde{\pi} = 0.9$ |
|---|---|---|---|
| RBF SVM($\gamma = 0.001, C = 1$) | 0.0452 | 0.1258 | 0.1231 |
| GMM 4C Tied-Full | 0.0311 | 0.0823 | 0.0972 |
| Fusion | 0.0300 | 0.0815 | 0.0938 |




## 8.6 Conclusions

Our final model is the GMM with 4 components and tied covariance, overall it performs well since it can achieve a good 0.0295 in the main application for the min DCF metric, and decent values for the unbalanced application, respectively 0.08 and 0.092. Overall, the similarity between validation and evaluation results suggests that the evaluation population is similar to the training population and the choices we made on our training sets proved effective also for the evaluation data.