Politecnico di Torino

Margoni Diego Federico s295483@studenti.polito.it

Introduction

In this report I'm going to analyse a dataset consisting in 12 continuous features representing embedding speaker samples. The training dataset has 6000 samples., while the evaluation dataset has 4000 samples both halved between male and female voices. Female samples are classified with class 1 while male samples with class 0. All the analysis will be performed using 4-fold cross validation. We'll evaluate the model given three different applications (prior probabilities) 0.5, 0.9 and 0.1 relatively to Positive Class just to check the robustness of the model on different application, anyway all final considerations will be applied on the application that weight equally both classes and assign the same unitary cost to false positive and false negative classified samples.

During feature analysis we might notice clustering due to the fact that samples are taken from different ages groups, we although don't have explicit information about this.

I'll apply the basic steps to approach this classification task, comparing different classification models, using the Python programming language supported by the following libraries: Numpy, Scipy, Matplotlib, Sklearn, Seaborn.

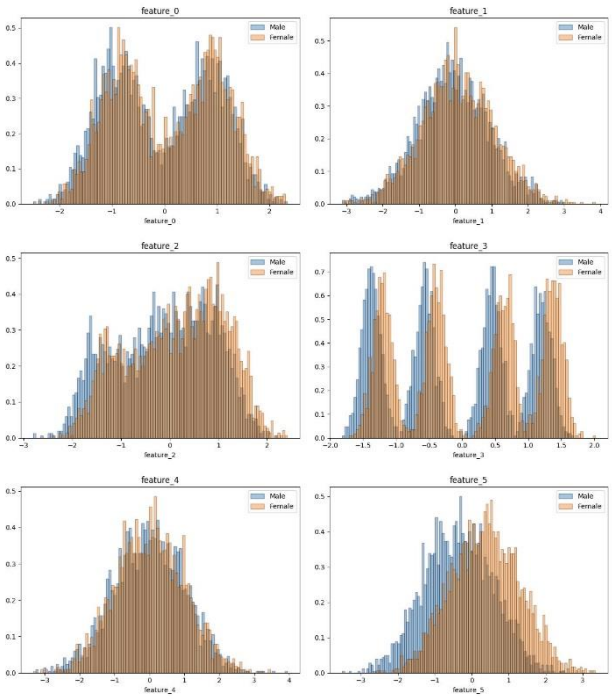
1. Features analysis

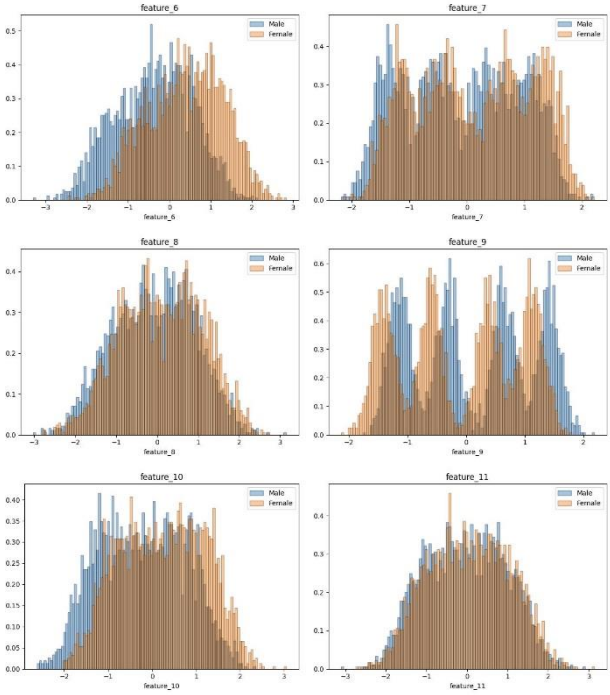
Before even analyzing feature distribution, we'll apply for the whole dataset and for all the following example, the Z-normalization on the features, to limit possible bias and have the same proportion for all the features:

$$\frac{x_i - \mu}{\sigma}$$

Following I'll evaluate if Gaussianization might be useful.

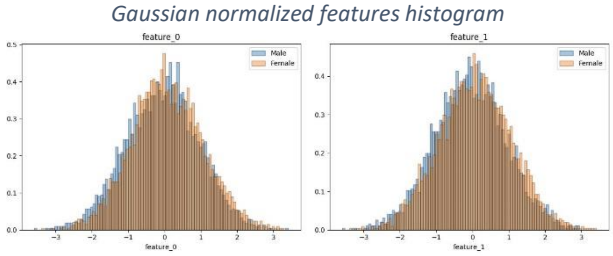
Z-normalized features histogram



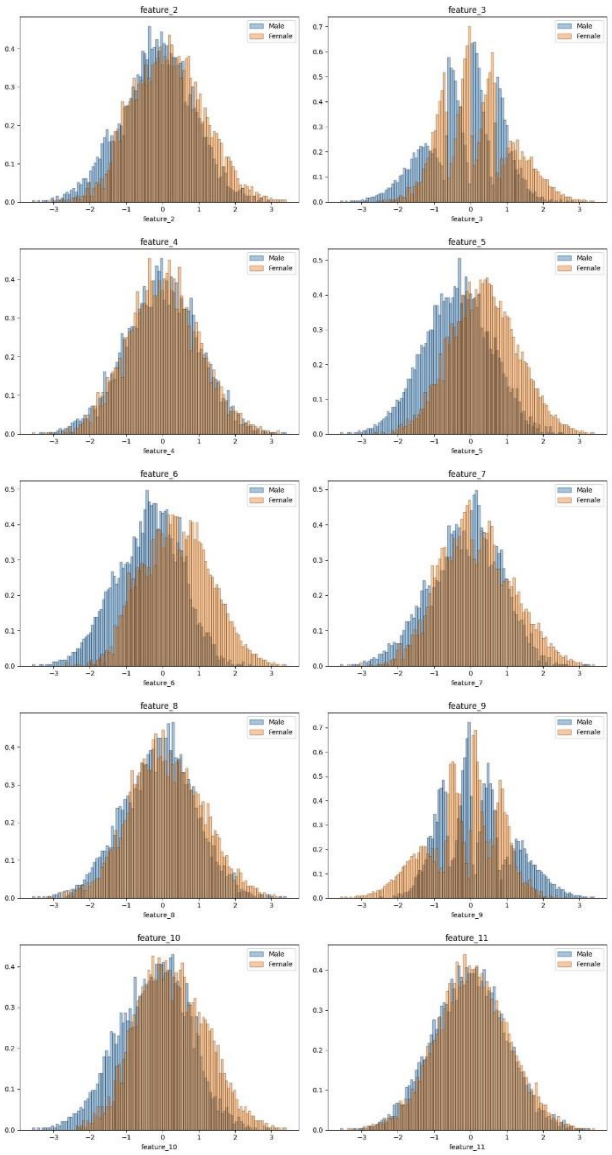


We can notice from training set features histogram that features don't show critical outliers, we also notice that the set of features composed by features 0 - 2 – 3 – 7 – 9 – 10 in particular are clearly not Gaussian-distributed, that's why we'll trying to apply Gaussianization and see if this will give benefits in particoluar for Gaussian models:

Gaussian normalized features histogram

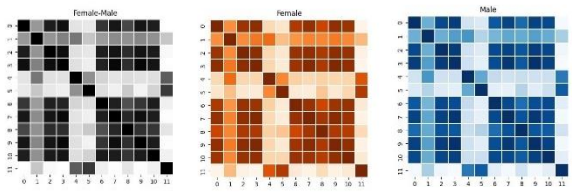


Margoni Diego Federico s295483@studenti.polito.it



Now we'll try to figure out features correlation through Pearson correlation coefficient:

Pearson correlation



We can immediately notice that most of the features are highly correlated, we might expect to have benefits from dimensionality reduction techniques like PCA (Principal Component Analysis) also considering that the number of features is quite high with respect to the number of samples.

2. Classification

2.a Gaussian Classifier

We'll start considering **Gaussian Classifiers**, in its three main variations, that are Full Covariance, Bayesian (diagonal covariance) and Tied Covariance. And both three cases will be analyzed with PCA dimensionality reduction till 3 features. We expect better result for full covariance and tied Gaussian classifiers, because it's very likely that samples have similar variation around the mean because we imagine that both female and male voices can vary in the same way. We don't expect good result from Diagonal Covariance because we'll lose all the correlations between the features.

	4-fold cross validation - min DCF						
	Z-Normalized				**Z and Gaussianized**		
	No PCA						
	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$	$\tilde{\pi} = 0.1$		$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$	$\tilde{\pi} = 0.1$
Full-Cov	0.049	0.123	0.133		0.065	0.194	0.196
Diag-Cov	0.566	0.863	0.827		0.530	0.850	0.822
Tied-Cov	0.047	0.125	0.125		0.063	0.194	0.193

	PCA (m = 11)						
	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$	$\tilde{\pi} = 0.1$		$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$	$\tilde{\pi} = 0.1$
Full-Cov	0.098	0.226	0.270		0.106	0.275	0.300
Diag-Cov	0.106	0.236	0.277		0.106	0.283	0.277
Tied-Cov	0.096	0.222	0.263		0.106	0.285	0.293

	PCA (m = 10)						
	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$	$\tilde{\pi} = 0.1$		$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$	$\tilde{\pi} = 0.1$
Full-Cov	0.114	0.262	0.307		0.111	0.278	0.300
Diag-Cov	0.118	0.284	0.302		0.111	0.268	0.269
Tied-Cov	0.111	0.260	0.301		0.110	0.275	0.292

	PCA (m = 9)						
	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$	$\tilde{\pi} = 0.1$		$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$	$\tilde{\pi} = 0.1$
Full-Cov	0.159	0.389	0.409		0.175	0.55	0.440
Diag-Cov	0.164	0.382	0.424		0.171	0.432	0.431
Tied-Cov	0.157	0.373	0.410		0.173	0.456	0.433

As we expected Diagonal model doesn't perform well in any case, we have better result for Tied model to indicate that the two classes Covariance can be well model by a unique within Covariance matrix of the two classes.

Unexpectedly applying PCA worsens performance in a heavy way for Z-Normalized features, probably all features are well discriminative for classes, it worsens softly Gaussianized features. In general, for the Gaussian models, we'll take in consideration only Z-Normalized feature with Tied Covariance Gaussian model. Gaussianized features are also more sensible to different application.

Margoni Diego Federico s295483@studenti.polito.it

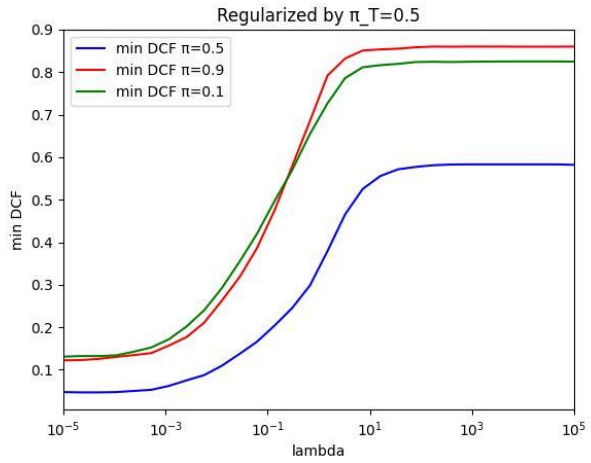
2.b Logistic Regression

Considering now discriminative model we try **Linear Logistic Regression** we'll starting tuning the regularization term λ for all three the applications, starting with the class-regularized version with the empirical prior $\pi_T = 0.5$. No PCA reduction will be applied.

From the figure below we will proceed for

$$\lambda = 10^{-5} \sim 0$$

Linear Logistic Regression lambda tuning with empirical prior 0.5



	4-fold cross validation - min DCF		
	Z-Normalized		
	No PCA		
	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$	$\tilde{\pi} = 0.1$
Log-Reg ($\lambda = 10^{-5}, not\ reg.$)	0.047	0.121	0.131
Log-Reg ($\lambda = 10^{-5}, \pi_T = 0.5$)	0.047	0.121	0.132
Log-Reg ($\lambda = 10^{-5}, \pi_T = 0.9$)	0.048	0.124	0.132
Log-Reg ($\lambda = 10^{-5}, \pi_T = 0.1$)	0.047	0.122	0.141

	4-fold cross validation - min DCF		
	Z and Gaussian		
	No PCA		
	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$	$\tilde{\pi} = 0.1$
Log-Reg ($\lambda = 10^{-5}, not\ reg.$)	0.56	0.153	0.165
Log-Reg ($\lambda = 10^{-5}, \pi_T = 0.5$)	0.56	0.154	0.164
Log-Reg ($\lambda = 10^{-5}, \pi_T = 0.9$)	0.57	0.154	0.157
Log-Reg ($\lambda = 10^{-5}, \pi_T = 0.1$)	0.56	0.173	0.177

From these results we can see that Logistic Regression is more agnostic on data distribution, that's why Gaussianized features have a slightly better result than Gaussian Classifier. We notice that different regularizations do not affect so much the result, probably because samples are balanced between the two classes.

Until now the best candidates are both Tied Gaussian Classifier and Linear Logistic Regression, always considering not Gaussianized features. Considering these pretty good results, we won't proceed in trying also the quadratic version of Logistic Regression.

2.c Support Vector Machine

We now will consider the **SVM model**, from now on we will drop considering both Gaussianized features and PCA. We'll start considering **linear** version where we will validate with the k-fold approach the best value for the hyper parameter C, considering for this evaluation a balanced version of the application. Since that the Primal solution is not differentiable, we will use the Dual formulation adapting it to the L- BFGS-B algorithm which can only handle the box constraint

$$0 \leq \alpha_i \leq C$$

and not the following one

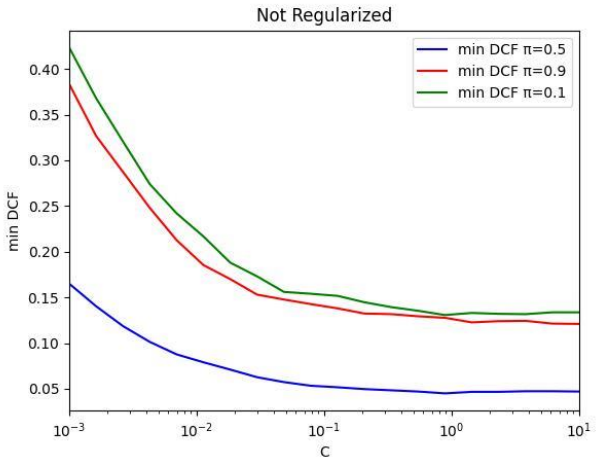
$$\sum_{i=1}^{n} \alpha_i z_i = 0$$

To drop the latter we will remove the bias term, from the primal formulation, and we will embedding it expanding the feature space of one dimension K; for this hyperparameter we will use the constant value 1 for performance and time consuming reason, even though choosing higher values for K would provide a solution closer to the original SVM.

From the figure below we will use:

$$C = 1$$

Linear SVM C Tuning



	4-fold cross validation - min DCF		
	Z-Normalized		
	No PCA		
	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$	$\tilde{\pi} = 0.1$
Linear SVM ($C = 1, not\ reg.$)	0.046	0.124	0.131
Linear SVM ($C = 1, \pi_T = 0.5$)	0.045	0.124	0.130
Linear SVM ($C = 1, \pi_T = 0.9$)	0.050	0.124	0.138
Linear SVM ($C = 1, \pi_T = 0.1$)	0.047	0.126	0.138

Margoni Diego Federico s295483@studenti.polito.it

We found another good candidate to be chosen as final classifier for our main application, it actually performs a little better that the Linear Logistic Regression.

Considering all the results we had so far, we can say that our dataset can be easily modelled by liner separation rules, so we could also not try the non-linear SVM kernel versions, but to have a stringer confirm of this supposition we will try the **SVM with polynomial (degree d=2) kernel.**

$$k(x_1, x_2) = (x_1^T x_2 + c)^d + \xi$$

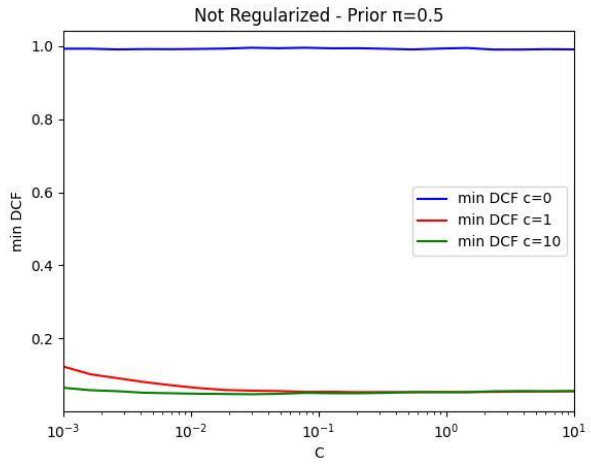
We will tune both the hyperparameter C and c. Unlike the Linear SVM we won't expand the feature space, because we'll compute the kernel on the original features (already expanded by the kernel formulation) and then add the bias

$$\xi = K^2$$

From the figure below we will use:

$$C = 1$$
$$c = 10$$

Polynomial kernel SVM C-c tuning



	4-fold cross validation - min DCF		
	Z-Normalized		
	No PCA		
	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$	$\tilde{\pi} = 0.1$
Poly SVM ($C = 1, not\ reg.$)	0.054	0.140	0.146
Poly SVM ($C = 1, c = 10,$ $\pi_T = 0.5$)	0.054	0.140	0.144
Poly SVM ($C = 1, c = 10,$ $\pi_T = 0.9$)	0.057	0.145	0.146
Poly SVM ($C = 1, c = 10,$ $\pi_T = 0.1$)	0.060	0.149	0.142

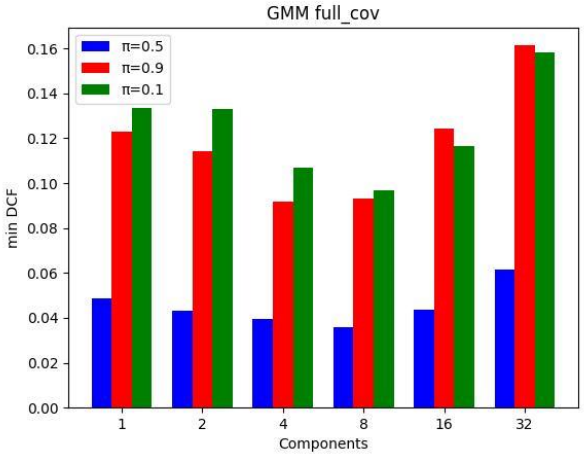
As expected the non-linear version of the SVM classifer with Polynomial kernel doesn't apport some particoular benefit to our task.

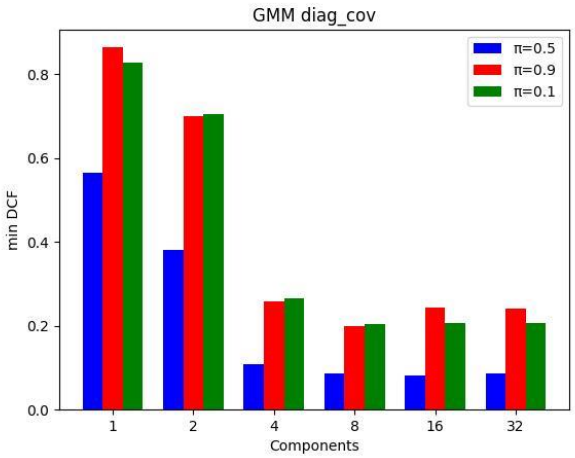
2.d Gaussian Mixture Model

Now we analyse the last classifier that is the **Gaussian Mixture Model,** we kind expect pretty good result from this classifier due to the natural propensity of this dataset to be split in sub-clusters as explained at the beginning, given by the different range of ages op people from whom the data was sampled. We'll test all three Gaussian model variation: Full Covariance, Diagonal Covariance and Tied Covariance (the latter limited to the components and not to the classes covariance).

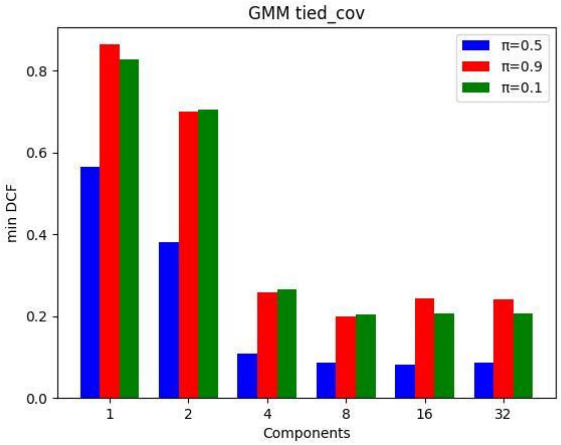
For this kind of model, we won't tune ant hyperparameter apart from the number of components, we'll use some empirical values for the following hyperparameters:

$$\Delta_t(stop\ criteria\ for\ the\ EL\ algorithm) = 1e-6$$
$$\psi\ (lower\ bound\ for\ the\ eighen\ values) = 0.01$$
$$\alpha = (displacement\ for\ the\ LBG\ splt) = 0.1$$





Margoni Diego Federico s295483@studenti.polito.it



	4-fold cross validation - min DCF		
	Z-Normalized		
	No PCA		
	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$	$\tilde{\pi} = 0.1$
GMM Full Cov (4 comp.)	0.039	0.092	0.107
GMM Full Cov (8 comp.)	0.036	0.093	0.097
GMM Full Cov (16 comp.)	0.044	0.124	0.116
GMM Full Cov (32 comp.)	0.062	0.161	0.158
GMM Diag. Cov (4 comp.)	0.108	0.257	0.265
GMM Diag. Cov (8 comp.)	0.086	0.199	0.205
GMM Diag. Cov (16 comp.)	0.082	0.242	0.208
GMM Diag. Cov (32 comp.)	0.085	0.240	0.207
GMM Tied Cov (4 comp.)	0.030	0.076	0.092
GMM Tied Cov (8 comp.)	0.030	0.074	0.090
GMM Tied Cov (16 comp.)	0.032	0.078	0.093
GMM Tied Cov (32 comp.)	0.032	0.077	0.103

The GMM model, in particular the Tied version, shows very good results for different ranges of components, we'll take as best candidate for our application the one with 4 components, both because it brings good result even for unbalanced applications and because it follows better the natural clustering of the dataset, so we suppose that, with respect to the one with 8 components, will perform better on the evaluation set.

We can conclude this phase comparing the best candidate's models so far.

	4-fold cross validation - min DCF		
	Z-Normalized		
	No PCA		
	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$	$\tilde{\pi} = 0.1$
Log-Reg ($\lambda = 10^{-5}, \pi_T = 0.5$)	0.047	0.121	0.132
Linear SVM ($C = 1, \pi_T = 0.5$)	0.045	0.124	0.130
GMM Tied Cov (4 comp.)	0.030	0.076	0.092

3. Actual Detection Cost Function

Up to now we have carried out our analyzes based on the min DCF, that is, for the binary case, as we were using an optimal treshold to perform class assignement, which can only be computed only if we already know the class of the sample we are classifying. Obviously, in the normal case we don't have this info, so we must make some additional considerations.

We can therefore use the Actual DCF, computing the theoretical threshold as a metric to perform class assignement:

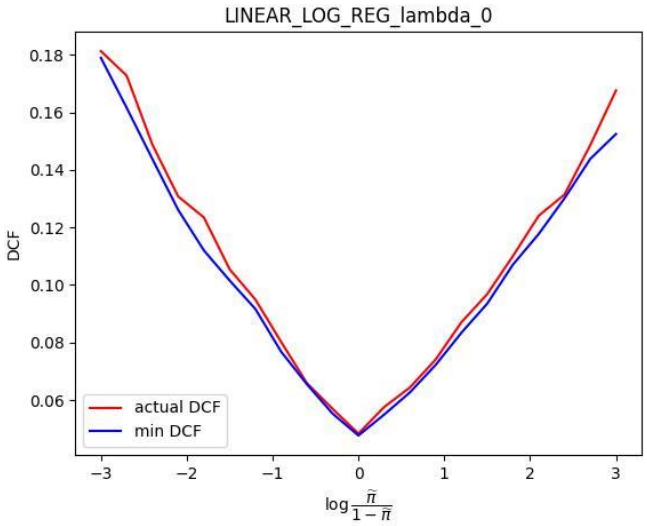
$$t = -\log\left(\frac{\tilde{\pi}}{1-\tilde{\pi}}\right)$$

To do this we must assume that scores are well calibrated, so we compute the Theoretical Treshold evaluating the Actual DCF to judge if we have a calibration loss and, in this case, we'll have to recalibrate the scores find a good treshold application-dependent hence for each application.

	min DCF – act DCF					
	Z-Normalized					
	No PCA					
	$\tilde{\pi} = 0.5$		$\tilde{\pi} = 0.9$		$\tilde{\pi} = 0.1$	
	min DCF	act DCF	min DCF	act DCF	min DCF	act DCF
Log-Reg ($\lambda = 10^{-5}, \pi_T = 0.5$)	0.047	0.048	0.121	0.133	0.132	0.137
Linear SVM ($C = 1, \pi_T = 0.5$)	0.045	0.046	0.124	0.281	0.130	0.269
GMM Tied Cov (4 comp.)	0.030	0.031	0.076	0.084	0.092	0.098

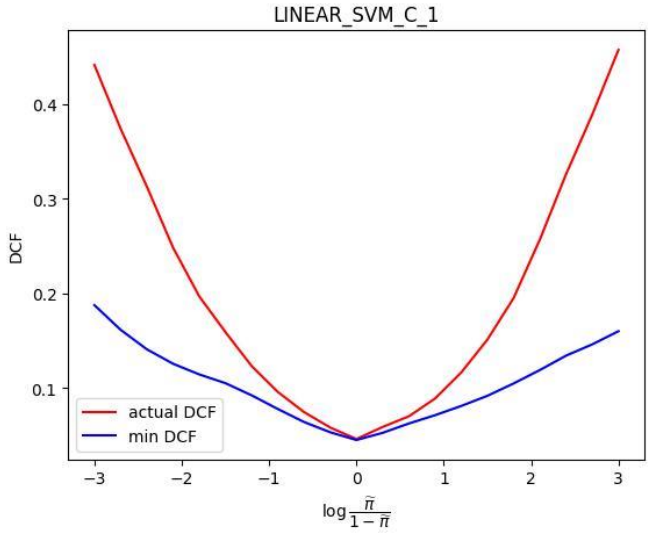
We can observe that the only model that shows some loss on varying of application is the linear SVM, the remaining classifiers provide a nice actual DCF that only gets slightly worse for extremely unbalanced applications. In the graph below we can see what was stated

Linear Logistic Regression Bayes errors Plot

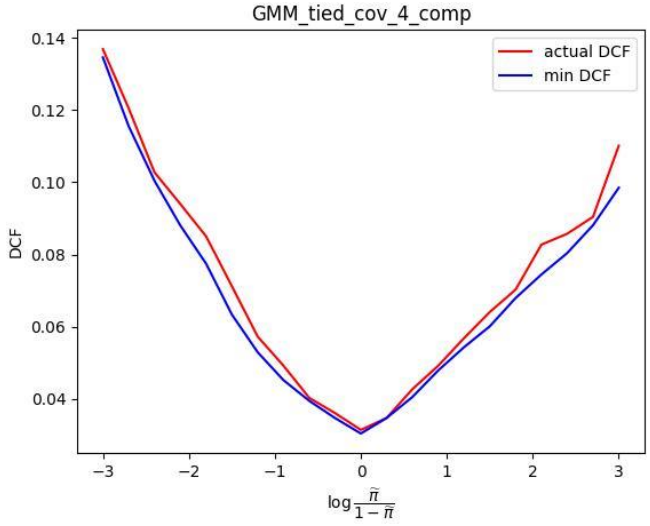


Margoni Diego Federico s295483@studenti.polito.it

Linear SVM Bayes errors Plot



Tied GMM Bayes errors Plot



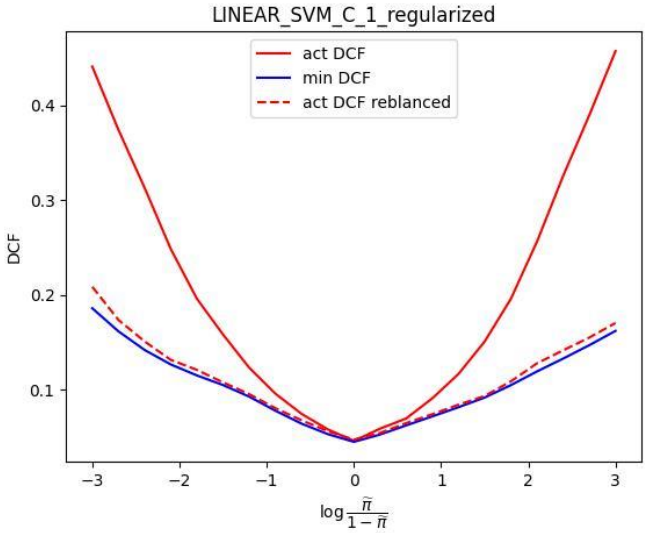
4. Score Calibration

As anticipated, we'll proceed, for the only Linear SVM classifier, to recalibrate the score. Considering the wide range of applications for whom the scores are mis-calibrated, instead of finding a good threshold application dependent we'll follow a second approach that consists in compute a transformation monotone function *f* that maps the classifiers scores to well-calibrated scores.

There are different approaches to deal with this approach, we will use the one that plans to train a new discriminative model of machine learning (i.e., prior-weighted Logistic Regression), directly on the scores provided by the classifier that we want to rebalance. Even in this case we will target a specific prior $\tilde{\pi}$, anyway the model will still provide good calibration for different applications.

So, we are going to calibrate scores for SVM Linear Classifier, using the computed score as input to a Logistic Regression Classifier (hyperparameter $\lambda = 0$) always using the 4-fold approach with application *prior = 0.5*, since the choice of the prior does not influence the overall performances.

Linear SVM Bayes errors Plot Calibrated



This plot shows how calibrating scores improved the classifier.

Our main choice remains that of GMM Tied with 4 components, that is the model that we will deliver. In the last part we will anyway try all three models over the evaluation dataset.

5. Experimental results

We'll start the evaluation on tests data considering minimum DCF metric, and then we'll evaluate using the theoretical threshold for all the classifiers.

	4-fold cross validation - min DCF		
	Z-Normalized		
	No PCA		
	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$	$\tilde{\pi} = 0.1$
Log-Reg ($\lambda = 10^{-5}, \pi_T = 0.5$)	0.0525	0.133	0.1355
Log-Reg ($\lambda = 10^{-5}, \pi_T = 0.9$)	0.053	0.1315	0.142
Log-Reg ($\lambda = 10^{-5}, \pi_T = 0.1$)	0.052	0.136	0.1385
Linear SVM ($C = 1, \pi_T = 0.5$)	0.051	0.131	0.251
Linear SVM ($C = 1, \pi_T = 0.9$)	0.051	0.127	0.1534
Linear SVM ($C = 1, \pi_T = 0.1$)	0.0525	0.1405	0.143
GMM Tied Cov (4 comp.)	0.0295	0.092	0.08

Margoni Diego Federico s295483@studenti.polito.it

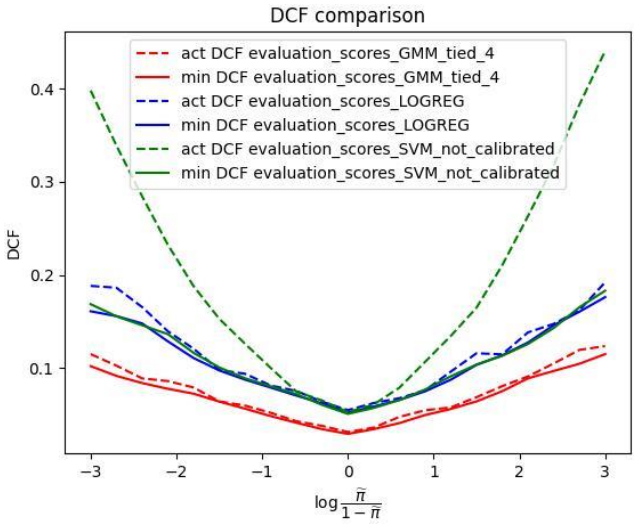
We can notice that the chosen model (GMM) provides result that are slightly better than the one we had during training. On the contrary the Log-Reg and Linear SVM got a bit worst result. In general, all three models have worst performance, with respect to the training phase, on the unbalanced applications.

We can now proceed on making evaluation not using the optimal threshold but using the theoretical one since scores are calibrated (we'll compare the SVM not calibrated scores with those after calibration). For the comparison we'll only consider our main application with *prior = 0.5*.

	min DCF – act DCF					
	Z-Normalized					
	No PCA					
	$\tilde{\pi} = 0.5$		$\tilde{\pi} = 0.9$		$\tilde{\pi} = 0.1$	
	min DCF	act DCF	min DCF	act DCF	min DCF	act DCF
Log-Reg ($\lambda = 10^{-5}, \pi_T = 0.5$)	0.0525	0.0549	0.133	0.1465	0.1355	0.1459
Linear SVM (not cal.) ($C = 1, \pi_T = 0.5$)	0.051	0.053	0.131	0.2795	0.141	0.251
Linear SVM (cal.) ($C = 1, \pi_T = 0.5$)	0.052	0.054	0.0515	0.054	0.0515	0.054
GMM Tied Cov (4 comp.)	0.0295	0.0315	0.092	0.096	0.08	0.088

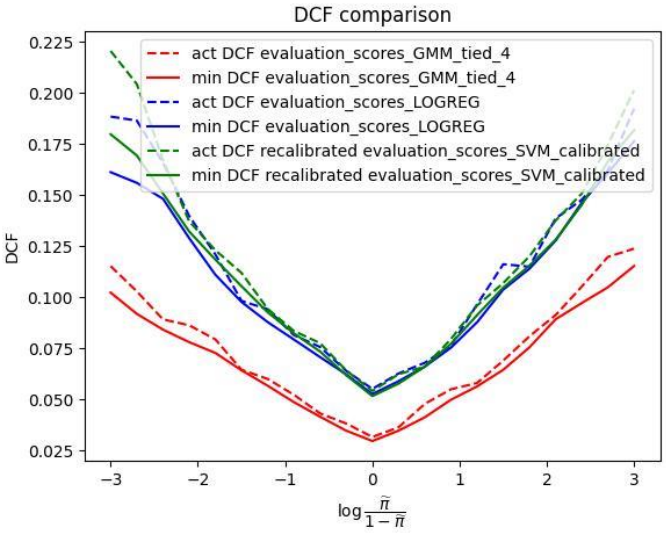
Scores from GMM are well calibrated, scores from Log-Reg should have probably been calibrated to obtain results closer to the optimal ones, especially for unbalanced applications. SVM benefits a lot from score calibration.

Bayes Error Plot of three classifiers (SVM not calibrated)



We can see also from this plot that Logistic Regression performed a little worse on evaluation data especially for unbalanced applications that disadvantages the True class.

Bayes Error Plot of three classifiers (SVM calibrated)



GMM confirms to be the best classifier for this kind of task, in general, the choices made during training were accurate also on test data.