Biometric identity verification

Problem Overview

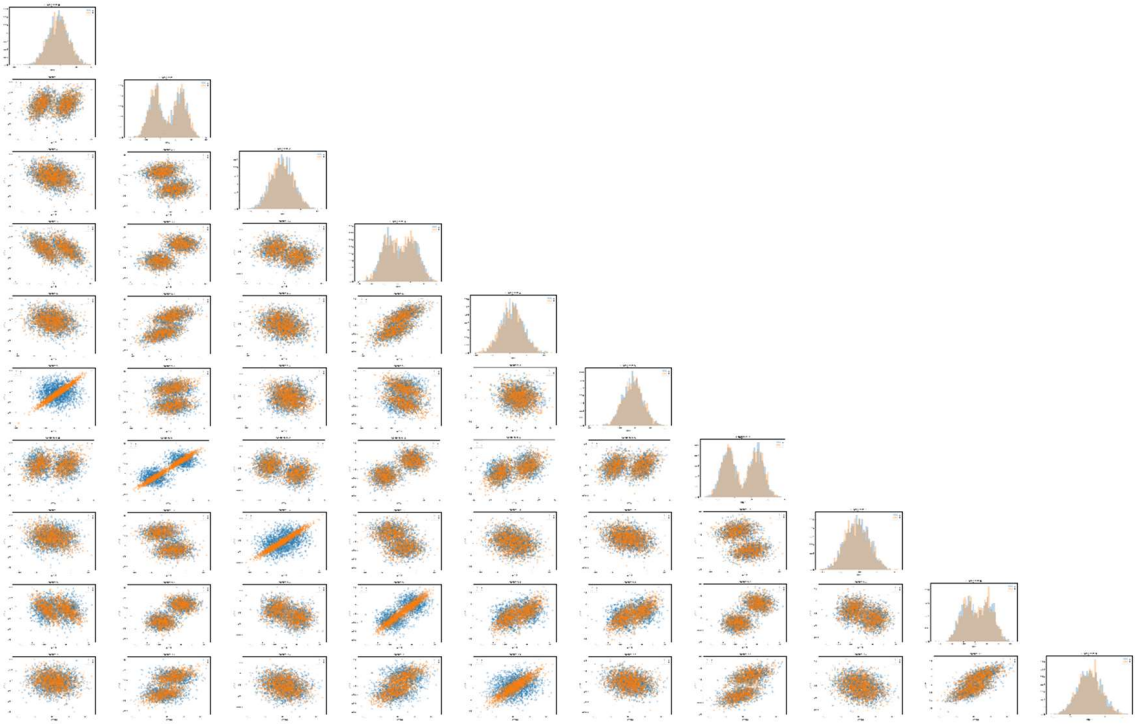
Determine whether the pair of embedding of speaker correspond to same (label 1) or different (label 0) speakers. The feature of each sample are 10, the dataset is composed of 2000 records with label 0 and 1500 with class 1. The desired application working point is (0.1,1,1).

We want to find a good classifier applying classic machine learning classification algorithms, dimensionality reduction techniques and the model evaluation procedure seen during course.

Data Overview

Raw data visualization:

In next scatter plot class 1 is orange and class 0 is blue.

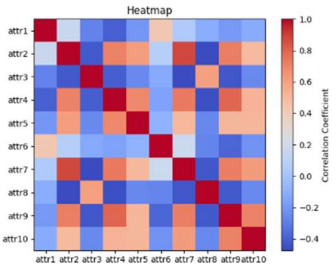


We see from histograms that data seems Gaussian distributed and can quite well see that in lot of cases, like explained in classroom there are 2 “clusters” in the same class records. We can also see a high correlation between features 0-5,1-6,2-7,3-8,4-9 in class 1 which mean that embeddings of the same person tend to have similar pair of features.

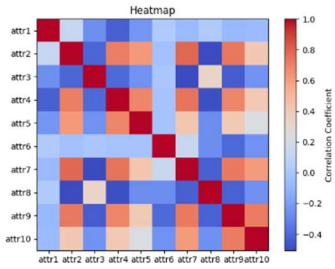
Federico Bussolino s317641, Francine Ombala s319110

Pearson's correlation coefficients:

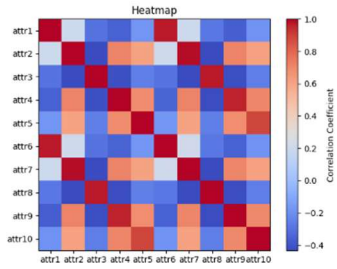
Dataset



Class 0



Class 1

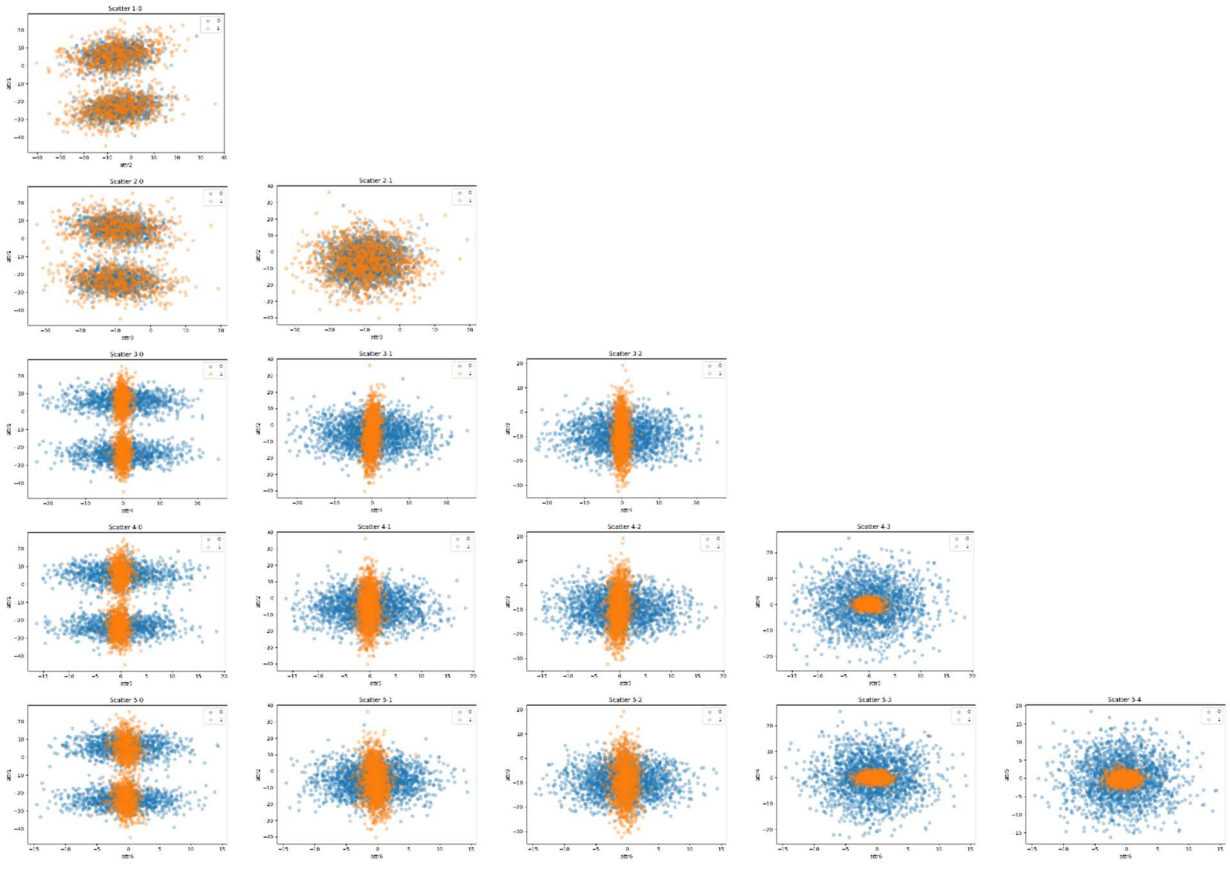


Per-class feature are highly correlated → we can't apply directly model that assume that features are not correlated. We can apply PCA keeping all the dimensions if we want only to remove feature correlations, but after that we need to verify that assumptions of the model still hold (ex. data are still gaussian distributed).

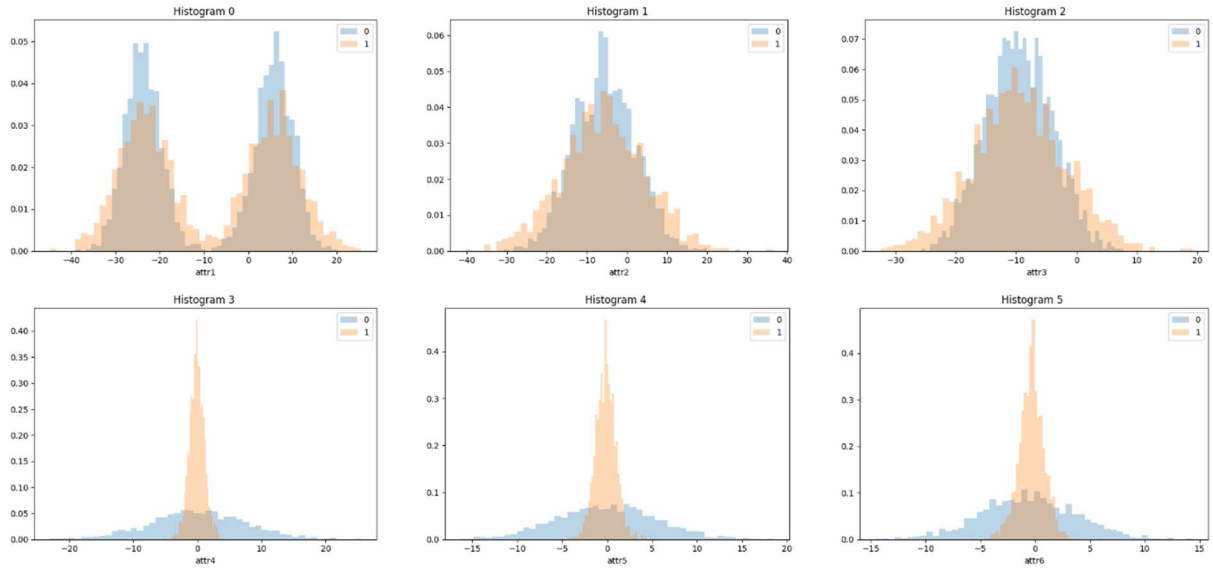
NOTE: remember that PCA diagonalize the overall covariance matrix, not the by-class covariance matrices, but we can hope it can remove the correlation as well.

PCA 6 data visualization:

Correspond to keeping an explained variance >95%.



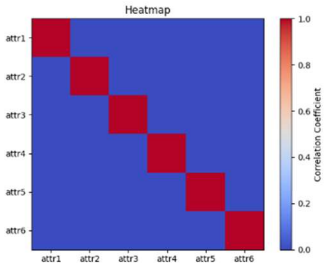
From those scatter plots we se that in lot of cases class 1 and class 0 can be separate quite well by ellipses.



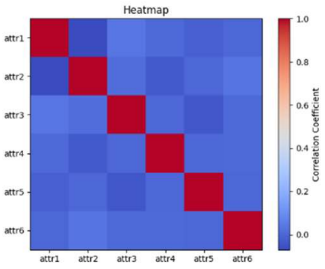
The histograms along new dimensions shows that if we consider data as modeled 1-d Gaussians, we can, thanks to dimension 3,4,5 separate well class 0 and 1. Moreover the obtained data distribution seems to be still a Gaussian after the linear transformation.

Pearson's correlation coefficients:

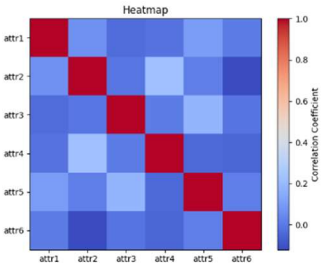
Dataset



Class 0



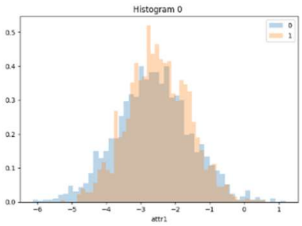
Class 1



We expect that a Naïve-Baies MVG can perform well since as assumption it treats dimensions independently, for class 1 assumption can be weaker since some correlation is still present.

I expect anyway that diagonal MVG can model well the data since correlation isn't really high.

LDA:



Since this is the only direction kept by LDA, we see that it is impossible to linearly separate the classes through tied covariance MVG.

More in general I think it's difficult to obtain good results on these data applying linear model since lot of feature couple shows in scatter plot are concentric ellipses.

Model selection process

We use k-fold cross validation with k=5 in order to have a reasonable time of training, particularly for SVM.

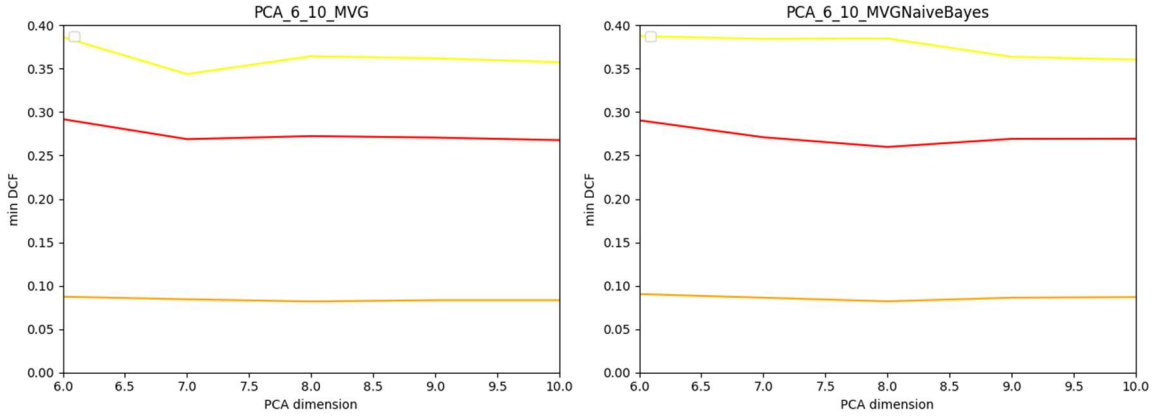
The performance evaluation of our model in this phase is based on minDCF.

Other 2 working points are evaluated: $\pi_1 = 0.5\ and\ 0.05$.

We choose 0.05 since we considered that a model that compare utterance to claim identity could be applied to security tasks, in which is particularly important to not give access to unauthorized users.
Since high cost of false positive correspond to lower prior of class 1, we reduce π_1 to 0.05.

Multivariate Gaussian Model

Consideration done during data overview are valid: data are gaussian distributed and features can be decorrelated through PCA, I expect classic and Naïve-Bayes to perform well, tied and Naïve-Bayes-tied will probably perform very bad since plotting data ellipses didn't have the same shape for different classes, moreover we've seen the LDA performed not good. I tried also PCA 5, but it throws away too much information.



Color map: red: $\pi_1 = 0.1$, orange: $\pi_1 = 0.5$ yellow: $\pi_1 = 0.05$

	Classic MVG			Naïve-Bayes MVG		
	$\pi_1 = 0.1$	$\pi_1 = 0.5$	$\pi_1 = 0.05$	$\pi_1 = 0.1$	$\pi_1 = 0.5$	$\pi_1 = 0.05$
NO PCA	0.268	0.083	0.358	--	--	--
PCA 6	0.292	0.087	0.386	0.291	0.090	0.388
PCA 7	0.269	0.084	0.344	0.271	0.086	0.384
PCA 8	0.272	0.082	0.364	0.260	0.082	0.385
PCA 9	0.271	0.083	0.362	0.269	0.086	0.364
PCA 10	0.268	0.083	0.358	0.269	0.087	0.361

Tied and diagonal-tied models performed very bad (minDCF=1.0 on PCA 6-10 data), as expected.

Best performing model for working point prior are highlighted in red and best for other working points are highlighted in blue.

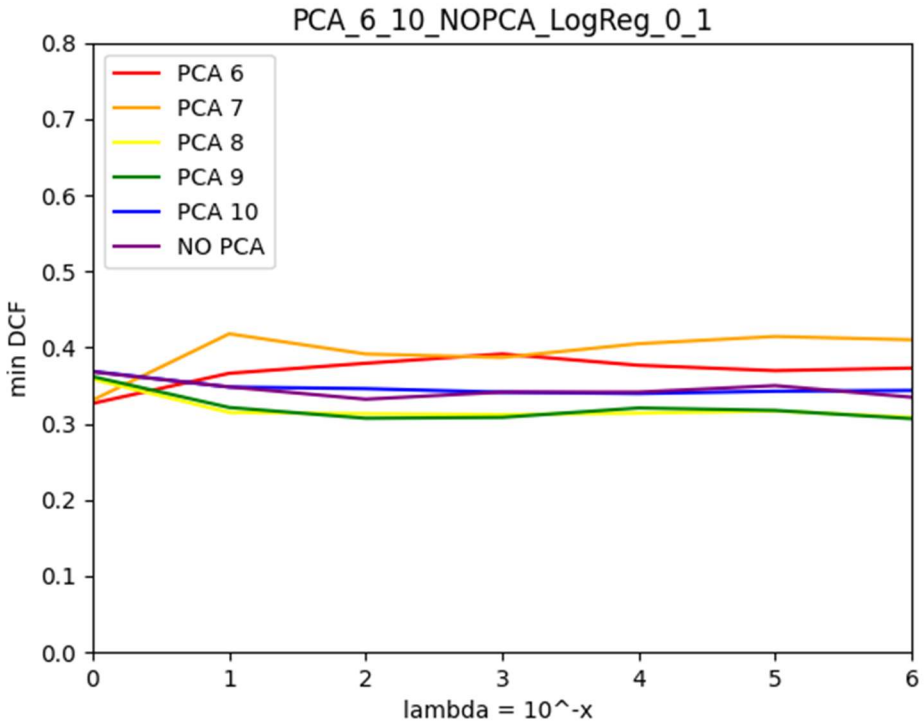
As we see the best tradeoff between information kept and overfitting reduction is achieved by PCA around 7-8, probably 7 works slightly better for classic MVG because it reduces overfitting due to the fact that it has more parameter.

Logistic Regression

Again, a linear classifier like Logistic Regression without expanded feature perform bad: for PCA in range 8-10 and $\lambda = 10^2, 10^1, 10^0, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$ minDCF at working point was always at 1.0.

Quadratic

I varied regularization term between 1.0 and 10^{-6} for PCA in range 6-10, obtaining following results:



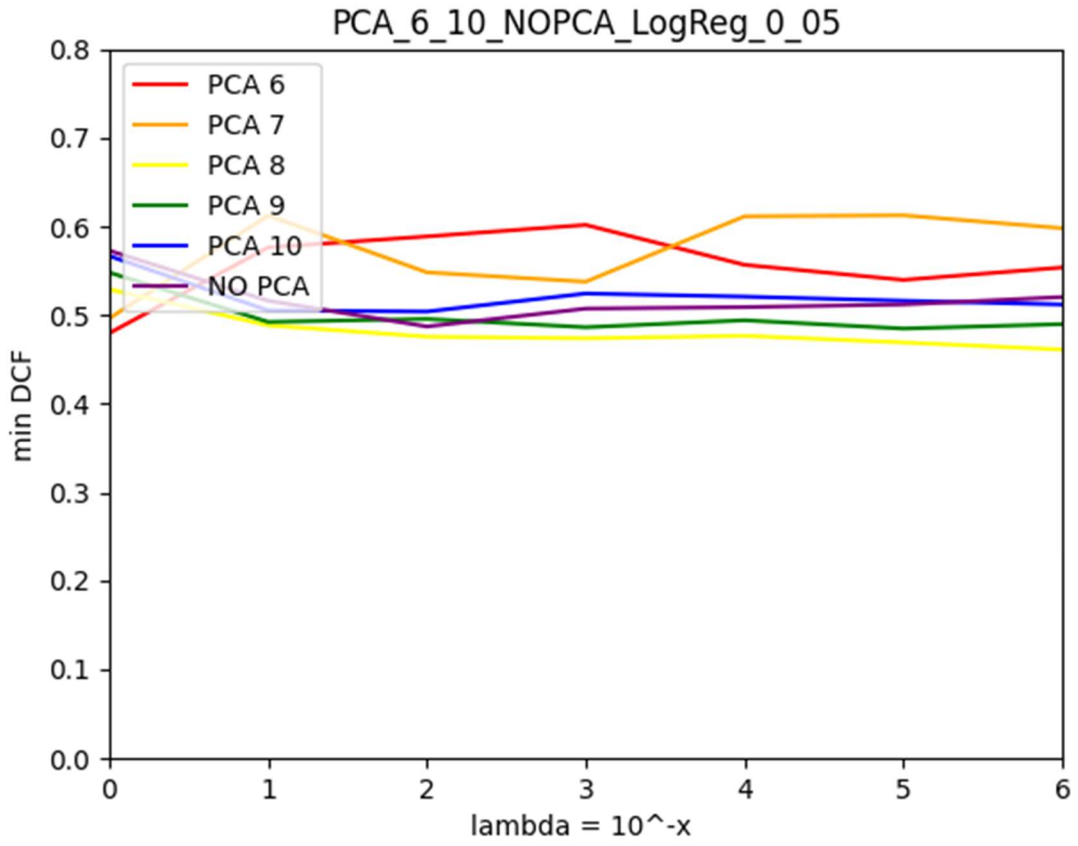
	$\pi_1 = 0.1$						
	$\lambda = 10^0$	$\lambda = 10^{-1}$	$\lambda = 10^{-2}$	$\lambda = 10^{-3}$	$\lambda = 10^{-4}$	$\lambda = 10^{-5}$	$\lambda = 10^{-6}$
PCA 6	0.327*	0.366	0.379	0.392	0.377	0.370	0.373
PCA 7	0.331	0.418	0.392	0.387	0.405	0.415	0.410
PCA 8	0.358	0.315	0.313	0.312	0.314	0.317	0.309**
PCA 9	0.362	0.321	0.307	0.308	0.321	0.318	0.307**
PCA 10	0.369	0.348	0.346	0.341	0.340	0.343	0.344
NO PCA	0.368	0.348	0.332	0.342	0.342	0.350	0.335

*Since I have seen that PCA 6 with high λ performed good I tried also $\lambda = 10\ and\ 10^2$. The resulting model is probably too simple: the costs are respectively 0.381 and 0.798.

** Since I have seen that PCA 8 and 9 with low λ performed good I tried also $\lambda = 10^{-7} and\ 10^{-8}$. The resulting model anyways perform slightly worse: 0.312 and 0.323 for PCA=8, 0.318 and 0.316 for PCA=9.

On other considered prior the model can achieve the following results. Since the costs are very similar I decided to don't show the plot.

	$\pi_1 = 0.5$						
	$\lambda = 10^0$	$\lambda = 10^{-1}$	$\lambda = 10^{-2}$	$\lambda = 10^{-3}$	$\lambda = 10^{-4}$	$\lambda = 10^{-5}$	$\lambda = 10^{-6}$
PCA 6	0.095	0.096	0.097	0.099	0.099	0.101	0.103
PCA 7	0.094	0.109	0.102	0.106	0.108	0.107	0.109
PCA 8	0.094	0.094	0.092	0.093	0.094	0.093	0.093
PCA 9	0.097	0.097	0.095	0.097	0.096	0.096	0.097
PCA 10	0.101	0.100	0.099	0.100	0.100	0.099	0.099
NO PCA	0.101	0.100	0.099	0.098	0.098	0.100	0.097



We can see even in this case that best performing model are generally those with PCA=8, but also those with PCA=9 behave quite well. I report also PCA=6 because for high lambda can also perform well.

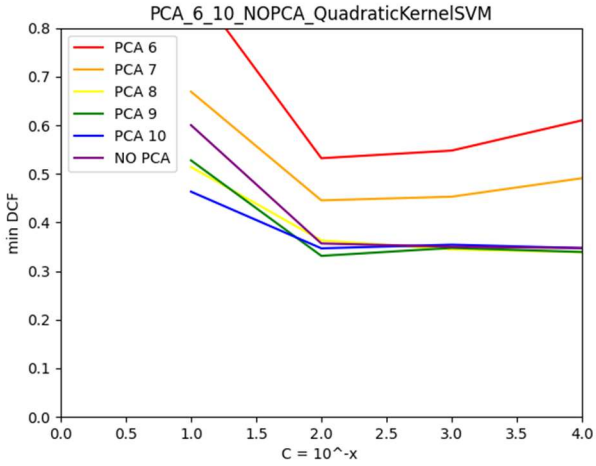
	$\pi_1 = 0.05$						
	$\lambda = 10^0$	$\lambda = 10^{-1}$	$\lambda = 10^{-2}$	$\lambda = 10^{-3}$	$\lambda = 10^{-4}$	$\lambda = 10^{-5}$	$\lambda = 10^{-6}$
PCA 6	0.480	0.576	0.589	0.602	0.557	0.540	0.554
PCA 8	0.529	0.489	0.476	0.474	0.477	0.469	0.461
PCA 9	0.548	0.492	0.496	0.486	0.494	0.485	0.490

We also tried Z-norm, but it is not very useful since features are already of same order of magnitude. This resulted in a minDCF of 0.522 for PCA=8 and lambda=0.01 and same value for NO PCA and same lambda.

SVM

Even in this case the linear version of SVM didn't have good performance.

Quadratic

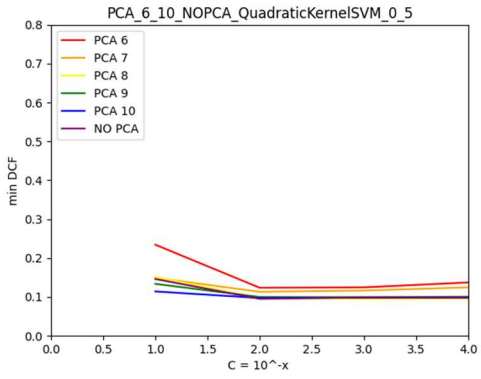


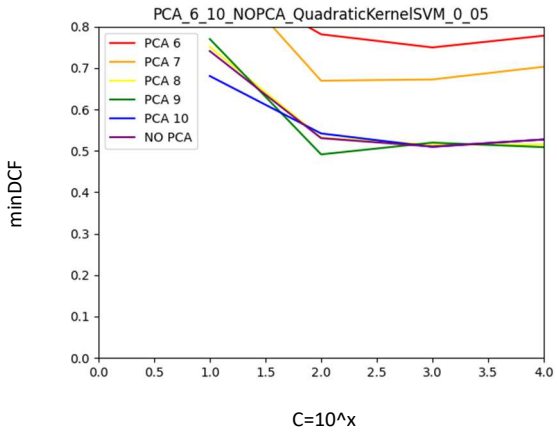
	$C = 10^{-1}$	$C = 10^{-2}$	$C = 10^{-3}$	$C = 10^{-4}$
PCA 6	0.896	0.532	0.548	0.610
PCA 7	0.669	0.446	0.453	0.491
PCA 8	0.514	0.363	0.345	0.339**
PCA 9	0.528	0.331	0.348	0.339**
PCA 10	0.463	0.347	0.354	0.347
NO PCA	0.600	0.357	0.350	0.347

I applied also $C = 10^0$ but it gives even worse results.

**I've tried also lower value of C for PCA 8-9, they don't give good results, in any case other models (Gaussians and LogReg) seems more promising.

For other priors error plots are the following:





The model that seems to perform better for 0.05 prior is PCA 9 with $C = 10^{-2}$ with a minDCF of 0.491. For the balanced prior (0.5) the best performing model seems to be the one without PCA and C=0.01 with a minDCF of 0.095. Similar values are obtained by applying same model to preprocessed with PCA 8 data. Meanwhile for the PCA 9 with $C = 10^{-2}$ we have a cost of 0.100.

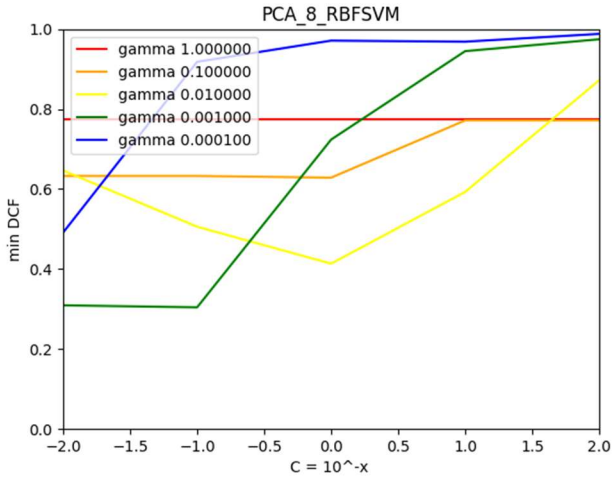
The fact that Quadratic SVM and Quadratic Logistic Regression differ so much in term of performance is a bit surprising for me.

RBF

I applied this model after PCA 8-9 since those are most promising number of dimensions.

PCA 8

Costs vary a lot based on γ and C:



	PCA 8, $\pi_1 = 0.1$				
	$\gamma = 10^0$	$\gamma = 10^{-1}$	$\gamma = 10^{-2}$	$\gamma = 10^{-3}$	$\gamma = 10^{-4}$
$C = 10^2$	0.775	0.633	0.646	0.309	0.491
$C = 10^1$	0.775	0.633	0.506	0.304	0.918
$C = 10^0$	0.775	0.629	0.414	0.724	x
$C = 10^{-1}$	0.775	0.772	0.593	0.945	x
$C = 10^{-2}$	0.775	0.772	0.872	0.974	x

I tried also C=1000 and $\gamma = 10^{-4}$ since I noticed a decreasing trend in cost→ I obtained a very low minDCF: 0.282, 0.086, 0.387. At higher cost minDCF return to increase probably due to overfitting. Data for low C and low γ are missing since I stopped the computation because I noticed an increasing trend in costs.

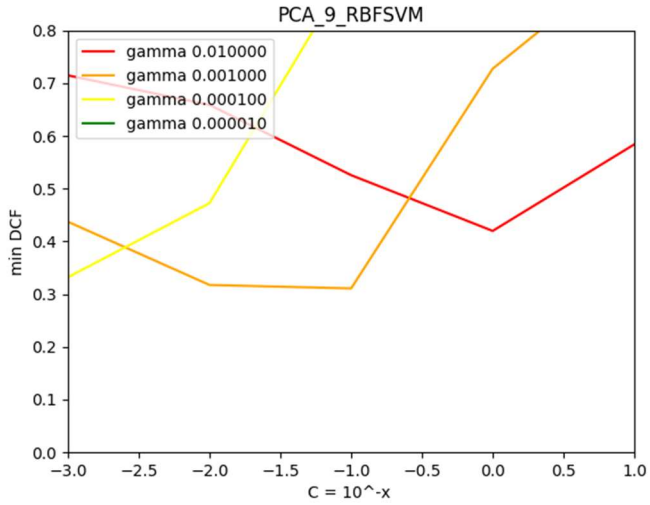
Seen previous results on the primary metric the only cost we are interested in are for prior 0.5 and 0.05:

	$\gamma = 10^{-3}, \pi_1 = 0.5$, *PCA 8*	$\gamma = 10^{-3}, \pi_1 = 0.05$, *PCA 8*
$C = 10^2$	0.095	0.469
$C = 10^1$	0.089	0.430

This application can perform better than quadratic logistic regression even with different priors.

PCA 9

Since with PCA 8 we notice that high costs are more effective we changed the range of C into $10^{3} - 10^{-1}$ and also $\gamma\ into\ 10^{-2} - 10^{-5}$.



Best results are those with $\gamma = 10^{-3}$:

	$C = 10^{3}$	$C = 10^{2}$	$C = 10^{1}$	$C = 10^{0}$	$C = 10^{-1}$
$\pi_1 = 0.1$	0.437	0.317	0.311	0.727	0.944

For other prior minDCFs are the following:

	$C = 10^{3}$	$C = 10^{2}$	$C = 10^{1}$	$C = 10^{0}$	$C = 10^{-1}$
$\pi_1 = 0.5$	0.110	0.096	0.088	0.193	0.571
$\pi_1 = 0.05$	0.604	0.435	0.441	0.818	0.980

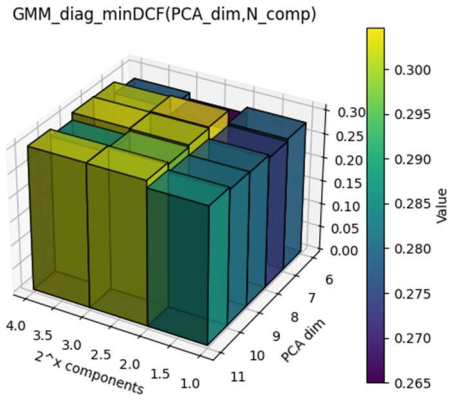
GMM

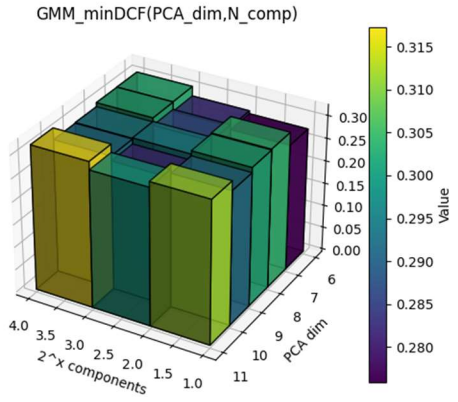
A 2 component GMM seem to fit well the data, I will also try 4 and 8 components at first for diagonal model (since they are computationally less expensive) with various value of PCA. If I notice some performance improvements I will try with more components even for other models (classic, tied and diagonal-tied). I don't expect a performance improvement since with more components we will probably have more complex model that can degenerate.

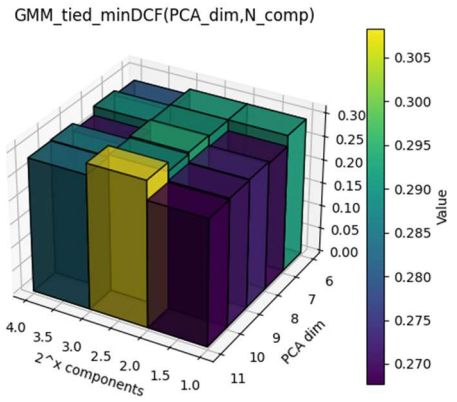
The tied version of the model can seem appropriate, in this case, since the 2 ellipses on scatter plot of PCA 6 for the same class are really similar, however a fully tied will probably fail like MVG tied.

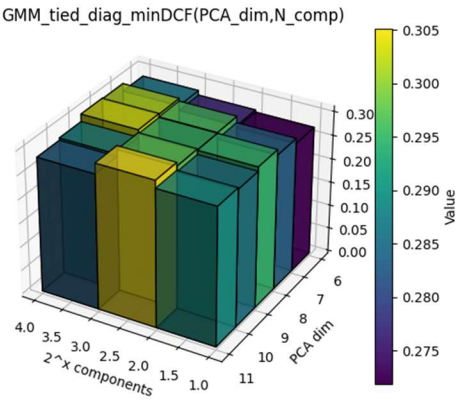
With diagonal model I will probably need PCA in order to decorrelate feature of components. Without PCA diagonal model didn't work well.

Plot of primary metrics based on number of components and PCA dimensions at given working point are the following:







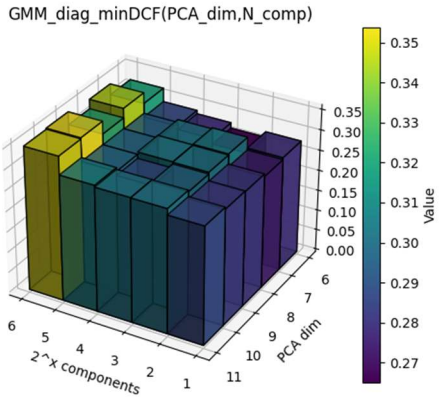


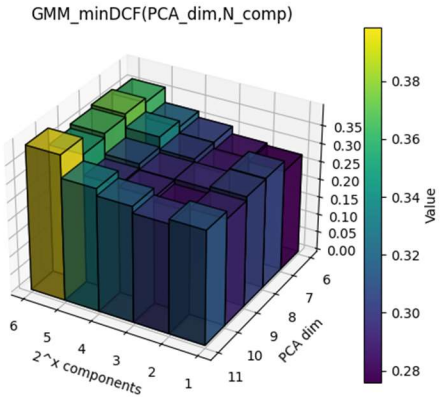
The model that best describes the distribution of our data can be chosen between the 2 component GMMs:

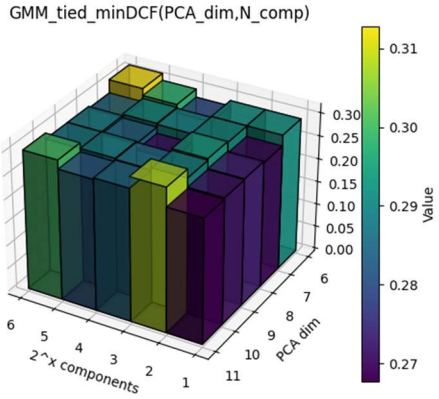
	$\pi = 0.1$				
	PCA 6	PCA 7	PCA 8	PCA 9	PCA 10
GMM	0.276	0.304	0.302	0.291	0.312
GMM diag	0.279	0.274	0.280	0.282	0.287
GMM tied	0.292	0.269	0.272	0.271	0.268
GMM diag tied	0.272	0.283	0.295	0.286	0.289

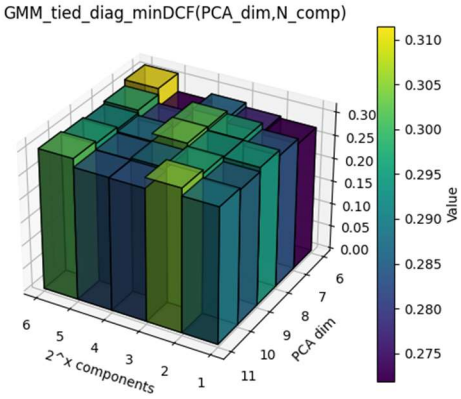
	$\pi = 0.5$				
	PCA 6	PCA 7	PCA 8	PCA 9	PCA 10
GMM tied	0.087	0.084	0.082	0.083	0.083
	$\pi = 0.05$				
	PCA 6	PCA 7	PCA 8	PCA 9	PCA 10
GMM tied	0.386	0.344	0.364	0.362	0.358

With larger number of cluster the model seems to perform a little worse (less noticeable for tied version):









In this case also diagonal model with 4 components, despite not being equivalent to 1d Naïve-Bayes performs well:

@WP 0.265, @0.5 0.093, @0.05 0.371

Since tied work well in general and gaussians of Class 0 are more "well separate" and uncorrelated (see heatmap PCA 6) I try a mixed model: tied for class 1 and diag-tied for class 0.

	$\pi = 0.1$				
	PCA 6	PCA 7	PCA 8	PCA 9	PCA 10
GMM Mixed	0.284	0.291	0.295	0.284	0.288
	$\pi = 0.5$				
	PCA 6	PCA 7	PCA 8	PCA 9	PCA 10
GMM Mixed	0.087	0.087	0.085	0.088	0.088
	$\pi = 0.05$				
	PCA 6	PCA 7	PCA 8	PCA 9	PCA 10
GMM Mixed	0.364	0.431	0.389	0.376	0.381

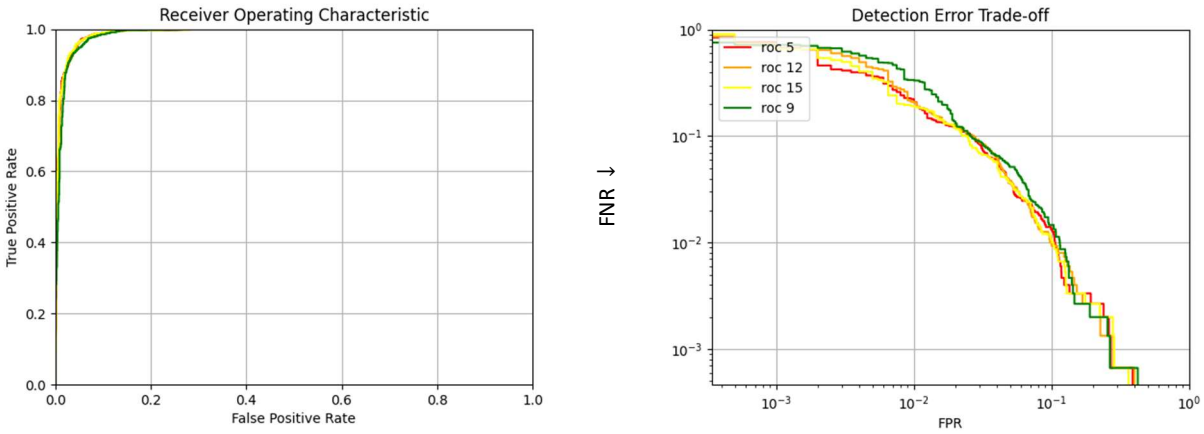
Since Naïve-Bayes MVG model had obtained good results may be worth try to model class 1 as a 1 component Gaussian and use tied for class 0 since tied performed better than non-tied.

Detailed analysis of best model

Best models (in term of minDCF) are:

Naïve-Bayes MVG (PCA 8)		
$\pi_1 = 0.1$	$\pi_1 = 0.5$	$\pi_1 = 0.05$
0.260	0.082	0.385
RBF kernel SVM (PCA 8, C=1000,$\gamma = 10^{-4}$)		
$\pi_1 = 0.1$	$\pi_1 = 0.5$	$\pi_1 = 0.05$
0.282	0.086	0.387
GMM tied (PCA 7, 2 comp)		
$\pi_1 = 0.1$	$\pi_1 = 0.5$	$\pi_1 = 0.05$
0.269	0.084	0.344
Quadratic LogReg (PCA 9, $\lambda = 10^{-2}$)		
$\pi_1 = 0.1$	$\pi_1 = 0.5$	$\pi_1 = 0.05$
0.307	0.095	0.496

We can see that Quadratic LogReg doesn't achieve sufficiently good performance since minDCF @ WP is higher than 0.3 (recommended actual cost).



Color map is: MVG, SVM, GMM, QLogReg.

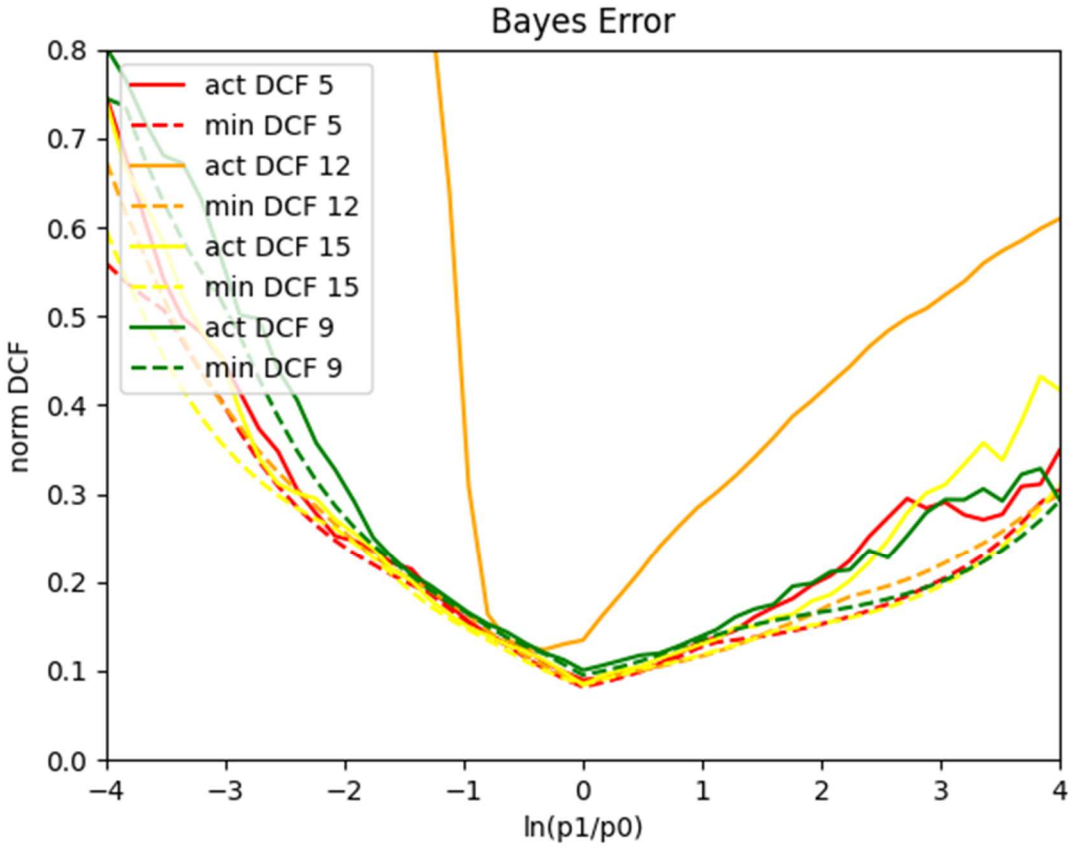
Plotting DET curve is more meaningful due to logarithmic scale (is still not the best scale to plot DET).

The portion of the graph we would take in account is both for x and y the range between $5 * 10^{-3}$ and $8 * 10^{-1}$, since out of this range out classifier will behave a lot like a dummy system:

In general MVG model perform good. They are particularly better at low FPR, also GMM perform well and in some range it outperforms MVG, in others it behaves really similar to SVM. Performance of Logistic Regression are worse in the considered range.

Comparison minDCF-actDCF

The considered prior 0.05 is @ -2.94 on the graph, so I preferred to plot it in range [-4,4].



Color map is: MVG, SVM, GMM, QLogReg.

As we expected SVM score were not well calibrated out of training range, anyways even if we calibrate we will, in the best case, end up with a minDCF that is similar to actDCF of MVG and GMM models. LR present even worse results.

We will try for calibration Linear LogReg on various λ→ We take in consideration Naïve-Bayes MVG (PCA 8) GMM tied (PCA 7, 2 comp) and their fusion.

Actual DCF after rebalancing becomes:

Naïve-Bayes MVG (PCA 8)		
$\pi_1 = 0.1$	$\pi_1 = 0.5$	$\pi_1 = 0.05$
0.273	0.090	0.427
GMM tied (PCA 7, 2 comp)		
$\pi_1 = 0.1$	$\pi_1 = 0.5$	$\pi_1 = 0.05$
0.285	0.084	0.422
Fusion (LogReg, $best\ \lambda = 10^{-2}$)		
$\pi_1 = 0.1$	$\pi_1 = 0.5$	$\pi_1 = 0.05$
0.266	0.085	0.459

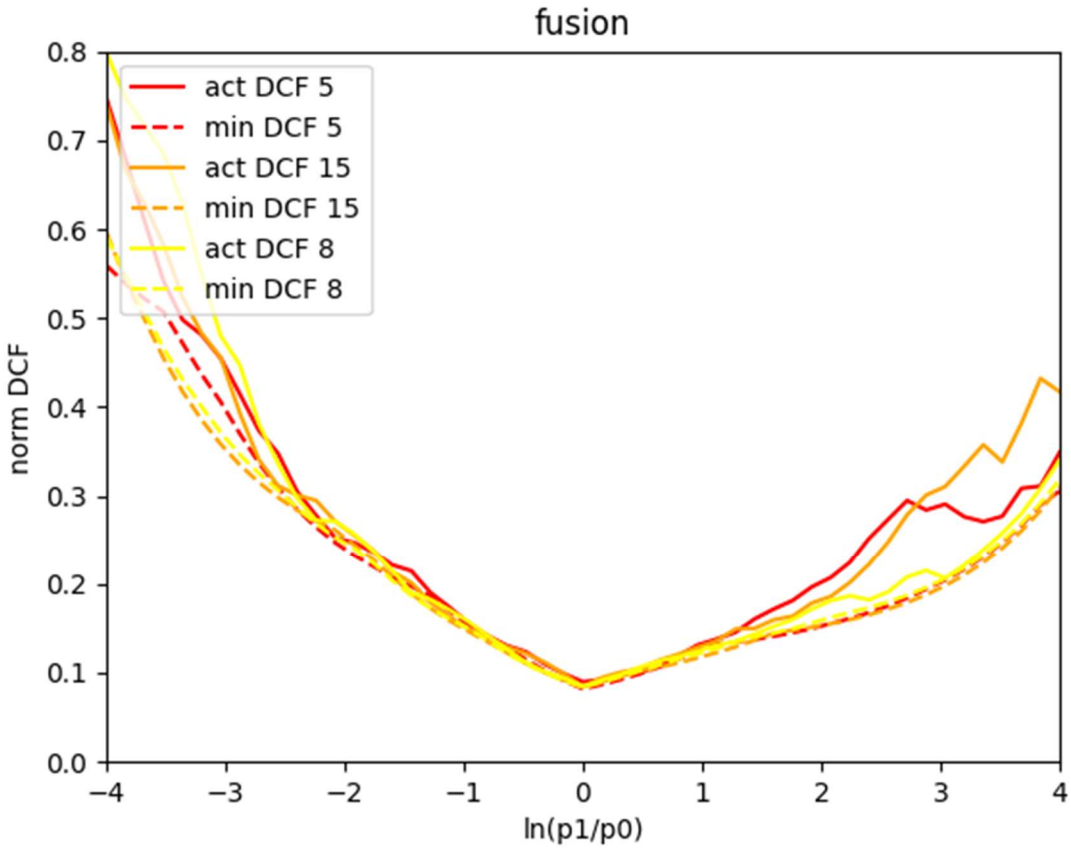
The previous fusion with GMM diag 4 with PCA 6 instead of GMM tied 2 gives similar results to the best already specified:

$\pi_1 = 0.1$	$\pi_1 = 0.5$	$\pi_1 = 0.05$
0.266	0.089	0.460

But if we look at the range approximatively of $\pi_1 = 0.1 - 0.15$. it is more similar to its minDCF.

I've applied also a fusion of 3 model: combination of (the model seen in last 2 fusions), but the results were not better.

We now look at a wider range of prior thanks to Bayes Error plot:



Color map is: MVG, GMM, Fusion.

We notice that we are in a lucky case since in correspondence of -2.2 (which is our working point) we have low cost → Fusion perform slightly better than the MVG model @WP.

Fusion is even more effective on unbalanced task with high π_1, but becomes less effective at very low π_1.

We choose the fusion model: LogReg($\lambda = 10^{-2}$) on (PCA7 ->2 component tied GMM, PCA 8->diagonal MVG).

Evaluation Results (test set)

Evaluation on test set resulted in higher cost for all most promising model.

The selected Gaussian model performed as follows:

actDCF	minDCF
fusion model LogReg($\lambda = 10^{-2}$) on (PCA7 ->2 component tied GMM, PCA 8->diagonal MVG)	
0.304	0.302
GMM tied 2 comp (PCA 7)	
0.308	0.305
MVG Naïve-Bayes (PCA 8)	
0.308	0.304
fusion model LogReg($\lambda = 10^{-2}$) on (PCA 6 ->4 component diag GMM, PCA 8 ->diagonal MVG)	
0.308	0.298

From this results we see that score of 2nd fusion model can provide better results with a different λ (ex. $\lambda = 5 * 10^{-2}$ gives an actDCF of 0.305).

Quadratic LogReg (PCA 9, $\lambda = 10^{-2}$)

actDCF	minDCF
0.871	0.737

RBF kernel SVM (PCA 8, C=1000,$\gamma = 10^{-4}$):

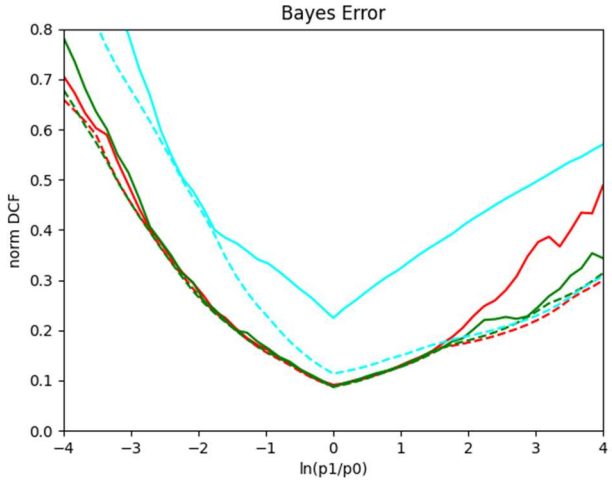
actDCF	minDCF
0.779	0.592

We can try to achieve better performance of Quadratic LogReg (PCA 9) by changing values of hyperparameters, anyways best results achieved are those with $\lambda = 10^{-2}$.

We can try to achieve better performance of RBF SVM(PCA 8) by changing values of hyperparameters, anyways as expected the best result is still distant from gaussian models:

γ: 0.001000, C: 100.000000 @WP→actDCF:0.499, minDCF:0.489

Through a Bayes error plot we can compare 3 interesting model at different priors:



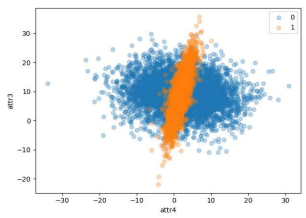
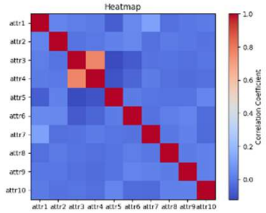
Color map: fusion 1(selected), fusion 2, SVM

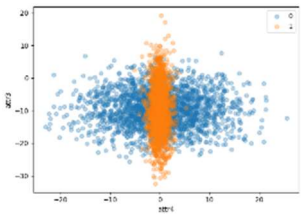
We can see that even if we apply score rebalancing on SVM model we could obtain a model that doesn't perform well as gaussian (in our range of interest, meanwhile for high π_1 it can perform better).

→ We can confirm that the choice of a gaussian model was optimal.

In order to understand why we didn't get cost < 0.3 we plot the scatter of features after PCA and the heatmap of class 1 after PCA.

→We see that despite PCA some correlations were not removed:

Test data→  

Train data→ 

Moreover plotting trainData and testData we see that there is a little difference in their distribution.

So we can use a model (2 comp classic GMM) that can capture correlations in order to capture correlations between features of original data (without PCA).

Applying 2 component GMM without PCA we obtained a better result:

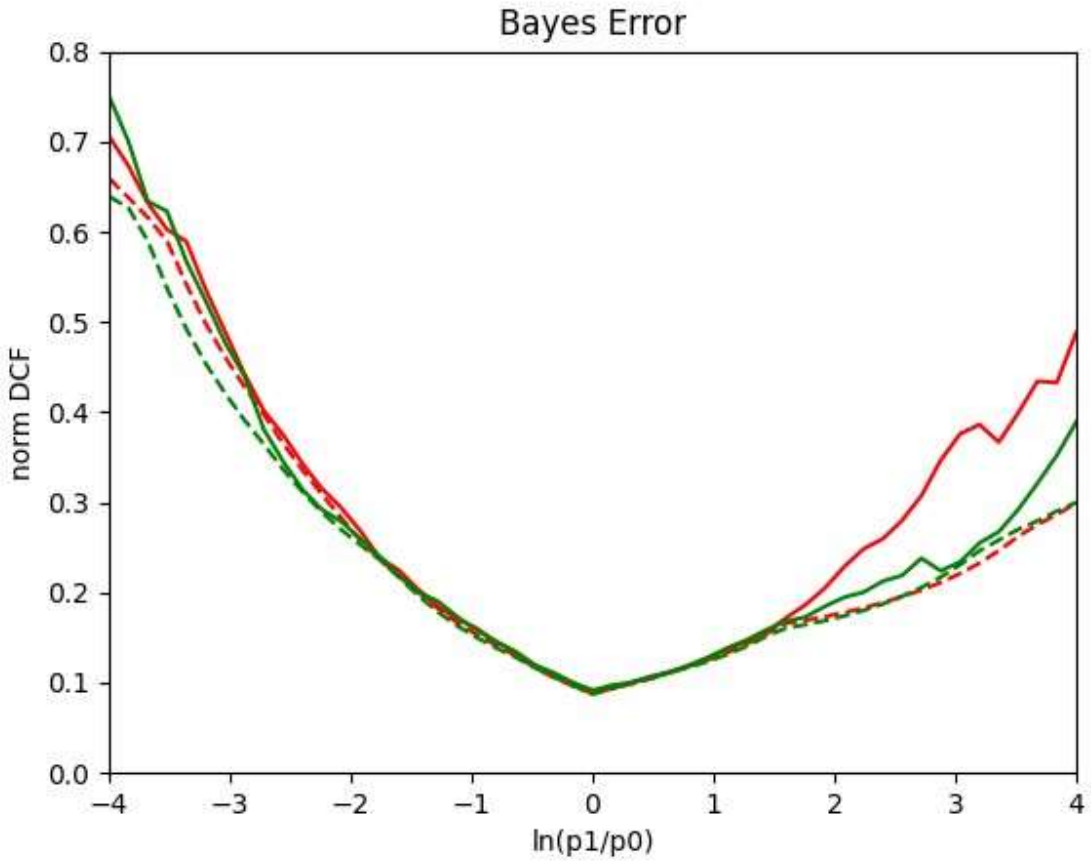
actDCF=0.297, minDCF=0.295.

In practice it was better only when it comes to use it on evaluation set. But in K-fold it gave worse results compared to fusion model.

The tied version without PCA performed even better:

actDCF=0.291, minDCF=0.282.

In conclusion I missed to try the GMM 2 components without PCA that in this case have better performance. The combination of this model with MVG Naïve-Bayes result less effective on test set than the model alone.

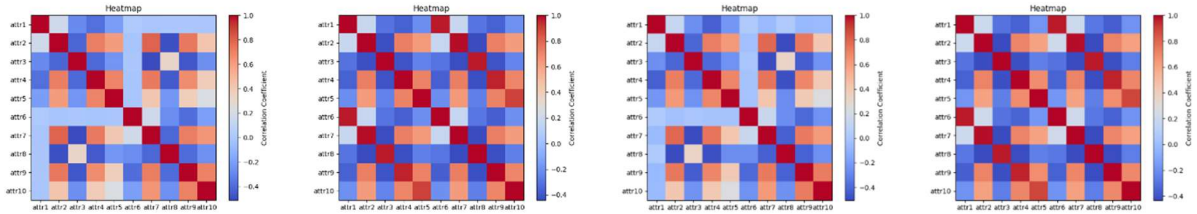


Color map: fusion 1(selected), GMM tied NO PCA 2 comp.

As we see in general a tied GMM with 2 components perform better achieving cost lower by 3% than the model we previously selected. We didn't gain too much cost due to miscalibration since we applied a gaussian model.

The fact that not diagonal model works better can be explained through the fact that correlation in the 2 sets are similar but the PCA transformed data correlations differ → the diagonal model didn't work as well as predicted.

Class 0 test set | Class 1 test set | Class 0 training set | Class 1 training set



As we see we could have obtained better result without PCA, but in general our approach and our assumption on the were good.