Fingerprint Spoofing Detection

Giuseppe Antonio Orlando

Contents

1 Task description **2**

2 Data Features **3**
2.1 Feature Distribution . . . 3
2.2 Pair of Data Features . . . 4
2.3 PCA . . . 4
2.4 LDA . . . 5

3 Generative Models **7**
3.1 Introduction . . . 7
3.2 Multivariate Gaussian Model . . . 7
3.3 Naive Bayes Assumption . . . 7
3.4 Tied Assumption . . . 8

4 Logistic Regression **9**
4.1 Introduction . . . 9
4.2 Linear Logistic Regression . . . 9
4.3 Qudratic Logistic Regression . . . 10

5 Support Vector Machines **12**
5.1 Introduction . . . 12
5.2 Linear SVM . . . 12
5.3 Kernel SVM . . . 14
5.3.1 Poly Kernel, degree 2 . . . 14
5.3.2 Poly Kernel, degree 3 . . . 15
5.3.3 RBF Kernel . . . 16
5.4 Rebalancing . . . 17

6 Gaussian Mixture Models **18**
6.1 Introduction . . . 18
6.2 Selection of components . . . 18
6.3 Diagonal or Tied . . . 18

7 Calibration **20**

8 Evaluation **22**
8.1 Quadratic Logistic Regression . . . 23
8.2 Poly Kernel SVM . . . 23
8.3 Gaussian Mixture Model . . . 24
8.4 Conclusions . . . 25

Chapter 1

Task description

The goal of this project is the development of a classifier to detect if a fingerprint image is authentic or spoofed (obtained artificially). The data that will be used in the classificatin task has a reduced number of features (10). In the following report label 0 will be used for class *spoofed fingerprints* and label 1 will be instead used for class *authentic fingerprint*. The target application working point is defined by the triplet $\pi_T = 0.5, C_{FN} = 1, C_{FP} = 10$.

We will start our discussion with an analysis of the dataset features. By doing this we will firstly check if there are some features that might help to distinguish the two classes. Dimensionality reduction techniques will also be considered.

After the analysis we will start building our classifiers using Multivariate Gaussians, Logistic Regression, Support Vector Machines and Gaussian Mixture Models. The best model will be chosen by the minDCF metric, evaluated with our application parameters. The cost will be evaluated also for different parameters to show how our classifiers perform on different applications.

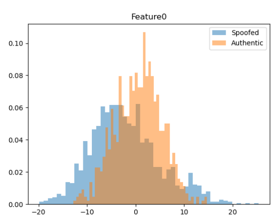
Once picked the models, we will finally perform calibration, if needed, and evaluation on our test set.

Chapter 2

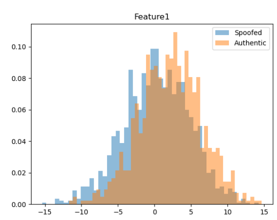
Data Features

2.1 Feature Distribution

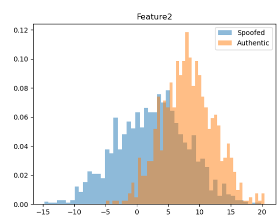
In the following pictures the feature distribution will be shown. The analyzed data has 10 features and the notation that will be used to enumerate them will start from 0.



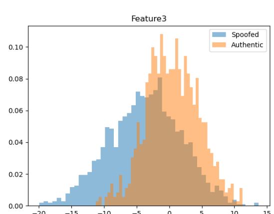
(a) Feature 0



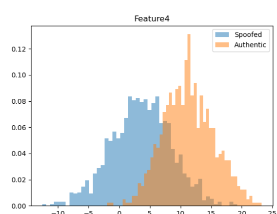
(b) Feature 1



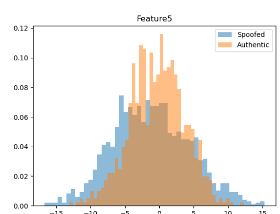
(c) Feature 2



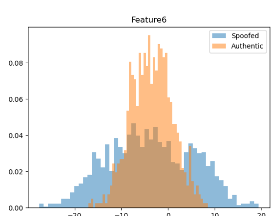
(d) Feature 3



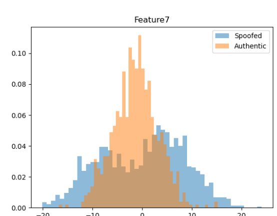
(e) Feature 4



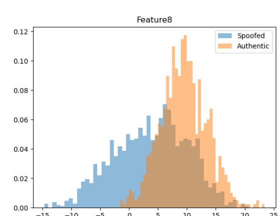
(f) Feature 5



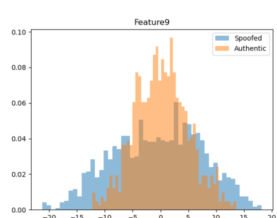
(g) Feature 6



(h) Feature 7



(i) Feature 8



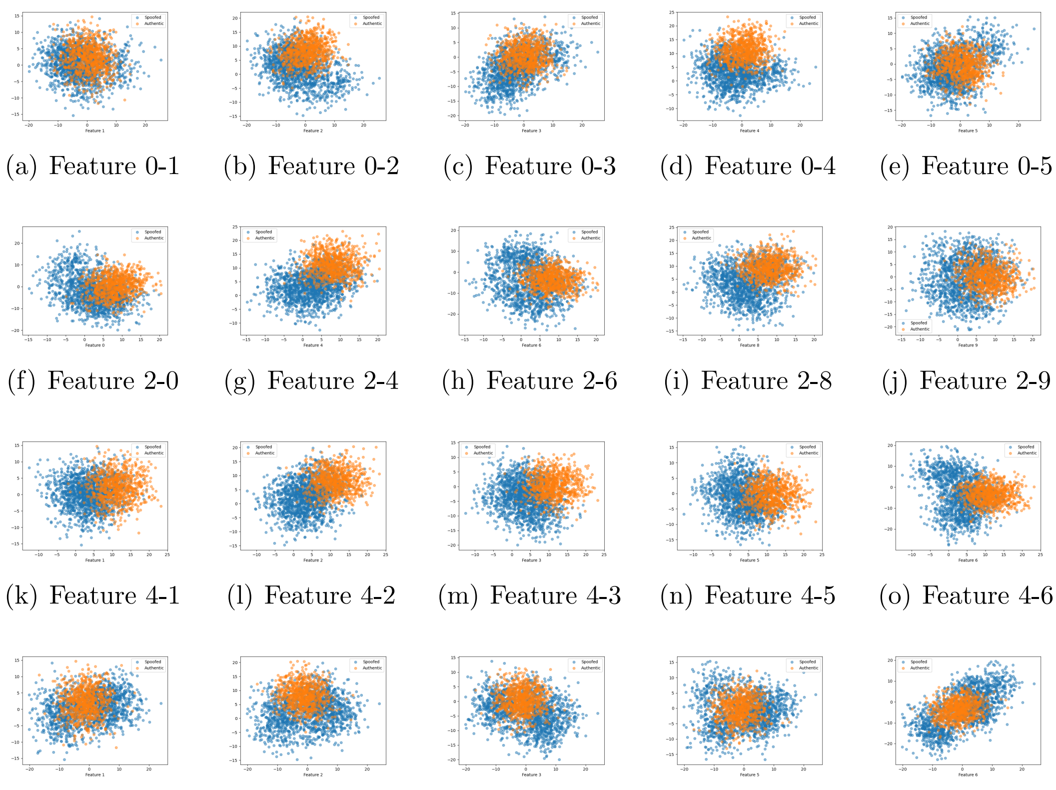
(j) Feature 9

Figure 2.1: Feature distribution

As we can see from the shown images, some features will not be useful in the classification task because their distribution in both classes overlap. For some features, like feature 2 and mostly in feature 4 a separation for the two classes is visible

2.2 Pair of Data Features

We will now compare the different features with each other to see if together they can create a good separation manner. The feature pair images will only be reported partially not to overflood the report.



(a) Feature 0-1 (b) Feature 0-2 (c) Feature 0-3 (d) Feature 0-4 (e) Feature 0-5

(f) Feature 2-0 (g) Feature 2-4 (h) Feature 2-6 (i) Feature 2-8 (j) Feature 2-9

(k) Feature 4-1 (l) Feature 4-2 (m) Feature 4-3 (n) Feature 4-5 (o) Feature 4-6

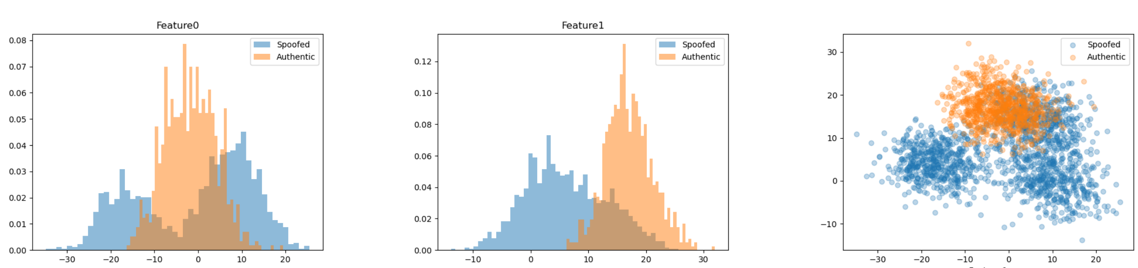
(p) Feature 7-1 (q) Feature 7-2 (r) Feature 7-3 (s) Feature 7-5 (t) Feature 7-6

Figure 2.2: Feature Pair distribution

The shown images confirm what was shown in the feature distribution histogram: feature 4 seems to be the feature that allows some sort of separion of the two classes.

2.3 PCA

Performin PCA on the data allows us to study and explore the Principal Components of the dataset, to capture the biggest variances. These are the histograms showing the distribution of the data along the two principal Components



(a) Principal Component 0 (b) Principal Component 1 (c) Pair Component 0 - 1

Figure 2.3: Feature distribution after PCA

From the shown plots it's possible to notice that the *Spoofed* class has a multiple cluster shape while the *Authentic* class has a more cohesive one. Looking at the PCA direction it looks like the features for class *Authentic* will be easly modelled with a Gaussian distribution, while the ones from the *Spoofed* class will not.

Computing the amount of variance captured by each PCA direction will also be usefull to understand how much dimentionality reduction will impact our classification task. In the following plot we will show how each component in pca sums up to 100% of variance.

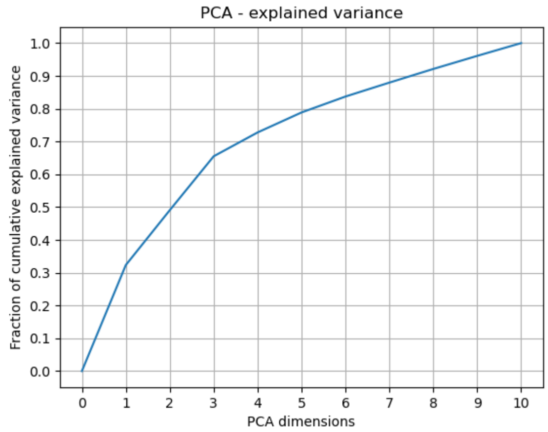
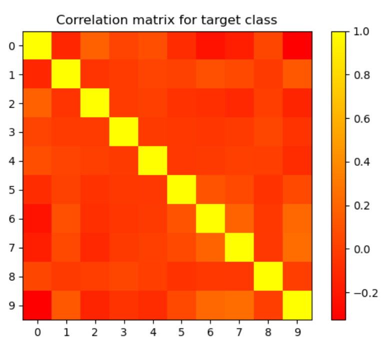


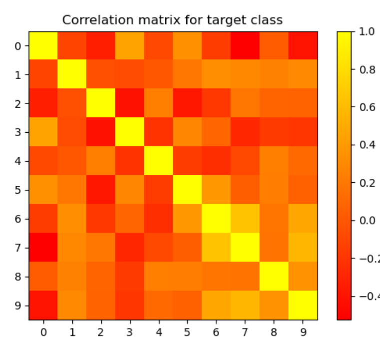
Figure 2.4: Explained variance in PCA

From the above plot, we can see that taking in consideration less than 6 PCA dimentions leads as to capture a variance of the dataset lower than 80% so important information about data might be lost. For this reason, whenever using PCA in the following report, we will consider range of values in the interval $[6, 10]$

Another great help given by PCA, is the removal of correlation between features. We will report the correlation matrixes for both non target and target class.



(a) Target Class



(b) Non Target Class

As it is possible to notice from the reported plots, the non target class shows high correlation between some features(like features 6 and 7); this correlation might worsen our classification task, so applying PCA can help removing the problem.

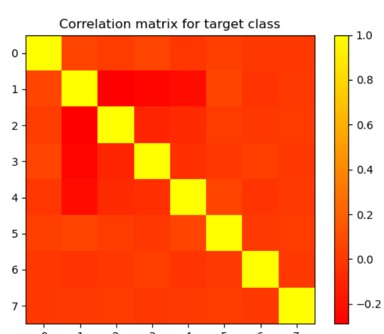


Figure 2.5: Non Target Class after PCA($m = 8$)

2.4 LDA

We will now perform Linear Discriminant Analysis to find the direction that provides the best split of our samples. The data distibution projected over that direction is represented in the following image.

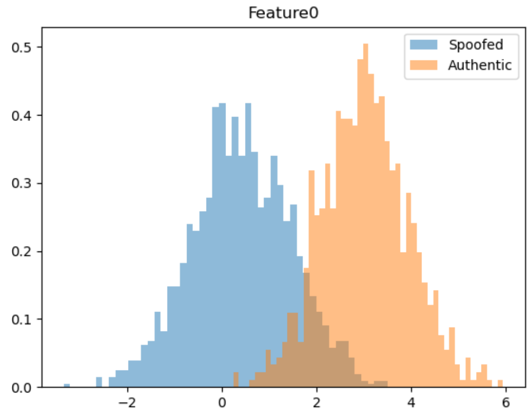


Figure 2.6: Data distribution on main LDA direction

As we can see in figure 2.6 a linear classifier might be doing a good job in dividing samples of the two classes. We can still see a quite expanse overlapping area that suggests, together with the previous scatter plots in figure 2.2, that higher order classifier might reduce the uncertainty for the data in the center.

Chapter 3

Generative Models

3.1 Introduction

The first approach in our classification task will be using generative models. We will try to obtain class posterior probability for each sample and with that we will decide which class it belongs to. For each model we will then consider the usage of PCA and will debate whether its usage has helped the performed task. For each of the models, in this and the following chapters, a *K-Fold* algorithm will be used with $K = 10$

Bayes cost will be used to compare the different models. For each model the DCF will be evaluated and, based on that, the models will be considered or discarded.

3.2 Multivariate Gaussian Model

In the following table we will show the results obtained training a multivariate gaussian model.

PCA dimentions	minDCF($\tilde{\pi} = 0.09$)	minDCF($\tilde{\pi} = 0.5$)	minDCF($\tilde{\pi} = 0.9$)
NO PCA	**0.333**	0.106	0.191
9	0.337	0.105	0.194
8	0.338	0.104	0.190
7	0.333	0.106	0.189

As we can see from the table above using PCA does not help that much in the classification task; the minimum DCF value for the target application is reached for $m = 7$ and $m = 10(noPCA)$.

3.3 Naive Bayes Assumption

We will now explore the results using the Naive Bayes assumption. The assumption is that the features are uncorrelated; as we have seen from the correlation matrixes this is more or less true for the target class, but the non target class presents some features that are highly correlated (0.7 correlation). This might create problems for the model in the classifications of objects of the *spoofed* class.

PCA dimentions	minDCF($\tilde{\pi} = 0.09$)	minDCF($\tilde{\pi} = 0.5$)	minDCF($\tilde{\pi} = 0.9$)
10	0.471	0.134	0.241
9	0.384	0.108	0.165
8	0.391	0.108	0.173
7	**0.391**	0.109	0.171

As expected the model performs worse than the non diagonal mvg model. The usage of PCA allows to reach better results because of the removal of correlated features, but they are still worse than the ones obtained with a regualt MVG model. For the other applications, the Naive Bayes assumption does not seem to worsen the classification task as it does for our target application. This might be due to our very high cost for a False Positive; the indipendence of features assumption does not hold in fact for the non target class and this might lead us to classify wrongly some spoofed fingerprints increasing as a consequence our minDCF.

3.4 Tied Assumption

A last attempt with gaussian generative models is using the tied assumption. The tied assumption generally helps applications with large dimentional data and small number of samples. In our case it is not strictly necessary but we will still apply the model and check it's performance.

PCA dimentions	minDCF($\tilde{\pi} = 0.09$)	minDCF($\tilde{\pi} = 0.5$)	minDCF($\tilde{\pi} = 0.9$)
10	0.472	0.171	0.378
9	**0.467**	0.170	0.391
8	0.473	0.167	0.391
7	0.467	0.168	0.394
6	0.474	0.170	0.393

From the table above it is pretty clear that the tied assumption does not help the classification problem. The Tied model, generates separation surfaces that are linear but, as we have seen from the LDA histograms, linear models will not work well on our dataset.

Chapter 4

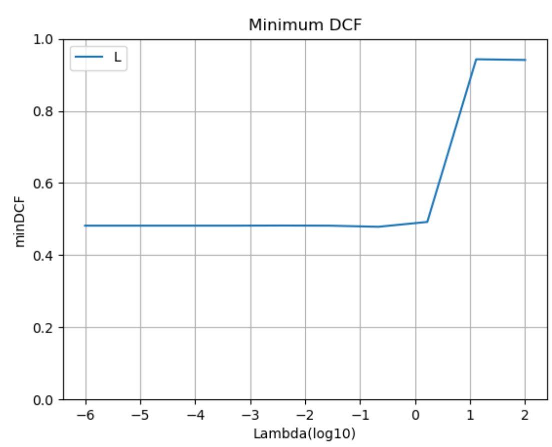
Logistic Regression

4.1 Introduction

In this chapter we will use Logistic Regression models for the classification tasks. Simple Logistic Regression computes separation surfaces that are linear, but as we have seen linear separation does not work well in our case. Better results are expected with a quadratic separation surface, so we will perform feature space expansion.

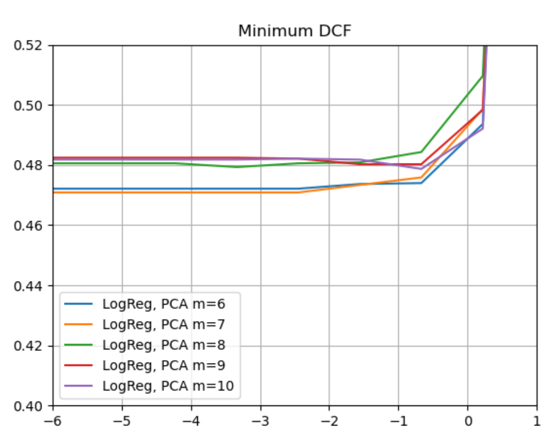
4.2 Linear Logistic Regression

The first step to perform is the selection of the hyperparameter λ, used in order to penalize solution with high norm of ω. After some some iterations these are the values of the minDCF reachable with different values of lambda.



As we can see from the picture above the most suitable values for the hyperparameter are the ones in the range $[10^{-6}, 10^{-2}]$. We will therefore use the value $\lambda = 10^{-3}$

From the results obtained in the previous chapter, we do not expect great results from the linear logistic regression because we have already noticed, in the tied approach, that linear classifiers actually perform worse than the non linear ones.



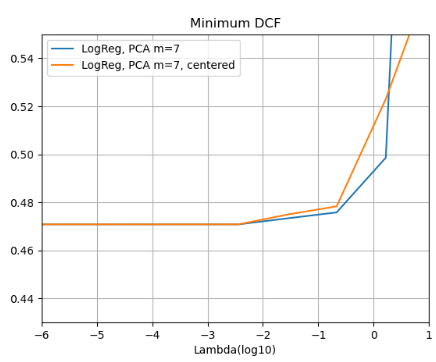
PCA dimentions	minDCF($\tilde{\pi} = 0.09$)
10	0.482
9	0.482
8	0.479
7	**0.471**
6	0.472

From the graph and the corresponding data reported in the table above, we notice that PCA helps to reduce the cost function. The best result is in fact reached with $PCA = 7$, but the cost is still significantly higher that the one obtained with quadratic generative models in the previous chapter.

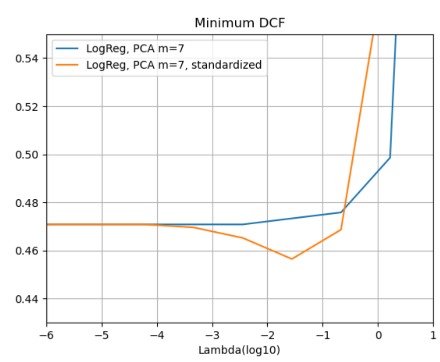
To improve the obtained results, it might be worth performing some data preprocessing before Logistic Regression, to check if we can improve the values of DCF obtained. These are all the strategies that will be adopted:

1. Centering of the data
2. Standardization of variances
3. Withening of the covariance matrixes
4. Length Normalization

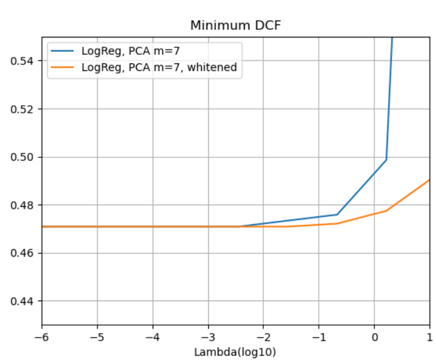
For each of the method above we will show the results for our best configuration ($m = 7$), and check if an improvement is detected.



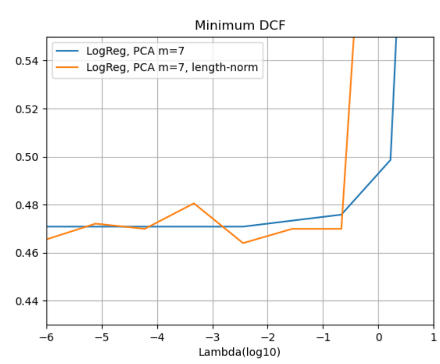
(a) Centerd Data



(b) Standardized Data



(c) Whitened Data



(d) Length Normalized Data

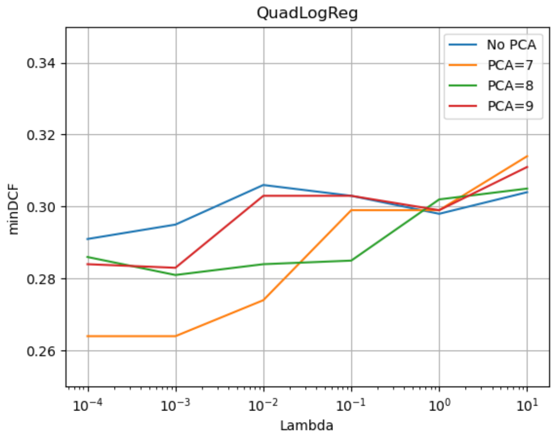
Standardization of the variances and length normalization both provide slightly better results, but the cost is still higher than quadratic models.

4.3 Qudratic Logistic Regression

Now we will try to expand the feature space in order to obtain, in the expanded space, a linear separation surface, that once mapped back into the original space will provide a quadratic separation. The mapping selected for the task is the following:

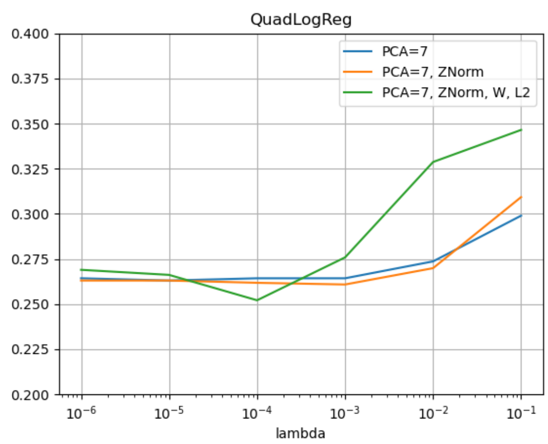
$$\Phi(x) = \begin{bmatrix} vec(xx^T) \\ x \end{bmatrix} \tag{4.1}$$

PCA will be performed before the data is mapped into the new space.



PCA dimentions	minDCF($\tilde{\pi} = 0.09$)
NO PCA	0.291
9	0.283
8	0.281
7	**0.264**

From the table we see that very good results are reached by expanding the feature space, better than the Gaussian Model. We will now try to apply some data preoprocessing to check if results can be improved even more. We will only consider data with PCA = 7 applied.



The results are somewhat similar for all three configurations but we can see that the minDCF (for our target) is reached for PCA = 7 and Z-Norm, Whitening, L2 applied to data before expansion.

PCA dimentions	Preoprocessing	minDCF($\tilde{\pi} = 0.09$)	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$
7	NO	0.263	0.094	0.206
7	Z-Norm	0.261	0.094	0.200
7	**Z-Norm + W + L2**	**0.252**	0.106	0.293

The chosen preprocessing helps the classification task for our application, but increases costs for the other two. One last thing to do is to try a prior weighted version of the algorithm. We will only do this for our best configuration that is: PCA = 7, Z-Norm, Whitenening and Length Normalization.

Prior	minDCF($\tilde{\pi} = 0.09$)	minDCF($\tilde{\pi} = 0.5$)	minDCF($\tilde{\pi} = 0.9$)
0.09	0.255	0.106	0.289
0.2	0.257	0.106	0.284
0.5	0.254	0.106	0.301
0.7	0.256	0.104	0.287

As we can see the prior weighted version of the algorithm does not improve performances.

Chapter 5

Support Vector Machines

5.1 Introduction

In this chapter we will make use of Support Vector Machines in order to classify data. Again we expect linear SVM not to perform well. We instead hope in good results using kernel SVMs

5.2 Linear SVM

To find the separation surface we will make use of the dual formulation of the problem. We will use cross validation in order to understand what is the best value for the hyperparameter C and also for the parameter K, employed to reduce the effect of regularization of the bias term b. Following is a graph which shows how the minimum DCF varies together with both C and K.

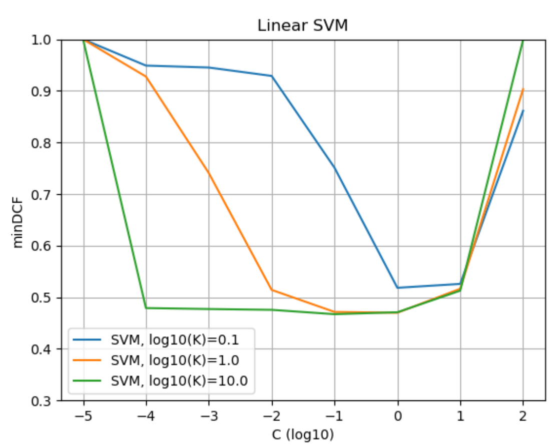


Figure 5.1: minDCF for various values of K and C

For the rest of the section we will use $K = 1$ in order to have easier computation. First we will analize the usage of PCA to see if it helps the classification task.

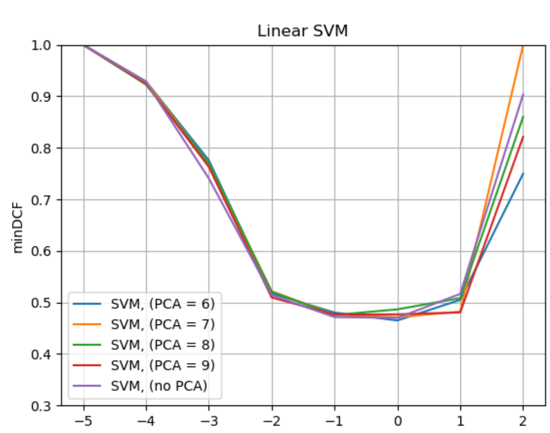
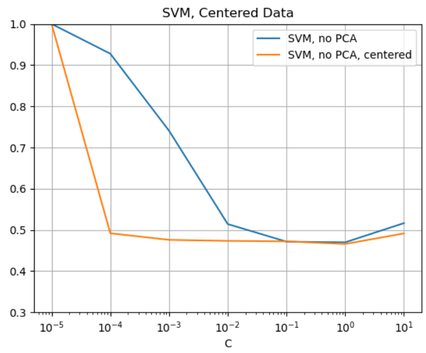
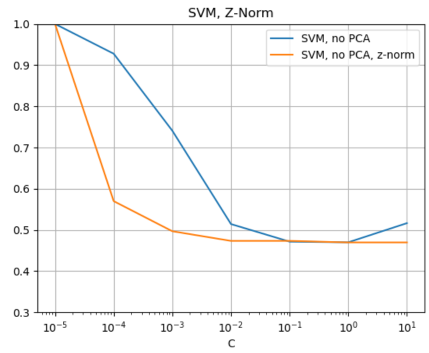


Figure 5.2: minDCF for various PCA values

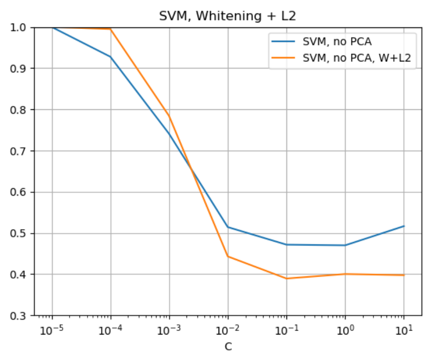
It's possible to notice that PCA does not help the classication task. From the graph we see that the minimum costs reached are in the range $[0.4, 0.5]$, much worse than our quadratic models. As a final attempt we will try different data preoprocessing strategies.



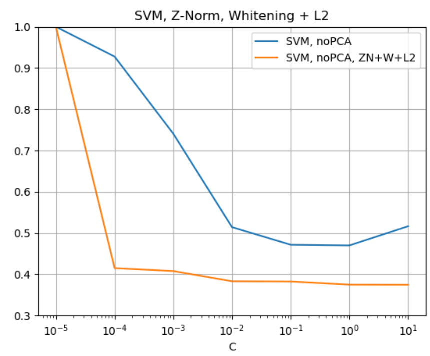
(a) Centered



(b) Centered with Std Variances

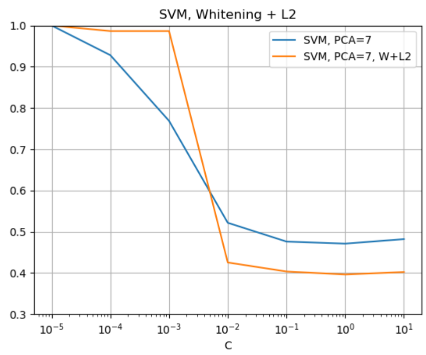


(c) Whitened and L2

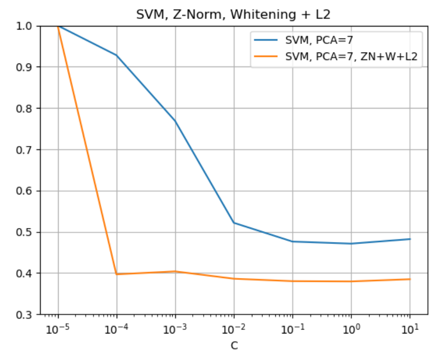


(d) Centered with Std Variances, Whitened and L2

Overall, all strategies seem to be helping the classification task but in praticular whitening lowers the minDCF significantly. Since all the data preprocessing strategies seem to be working we will try and apply them while working with $PCA = 7$.



(e) PCA, Whitened and L2



(f) PCA, Centered with Std Variances, Whitened and L2

We can notice how, even after data preprocessing and PCA, the results reached with this linear kernel are worse than the ones obtained with MVG or Quadratics Logistic Regression. As a consequence we will move to kernel SVMs and discard the linear ones.

5.3 Kernel SVM

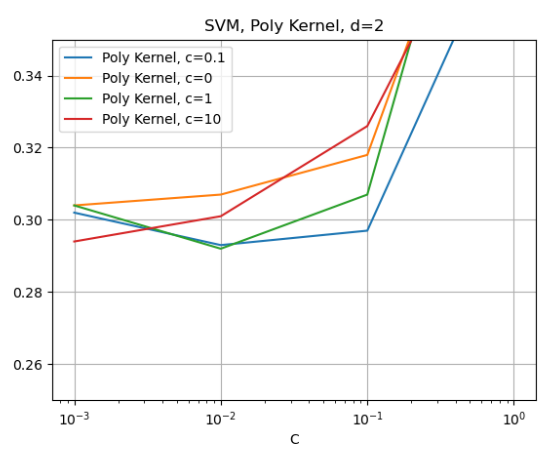
Two types of kernels will be considered in this section: Poly Kernels and RBF Kernels. For the poly kernels we will first analyze the performances of a degree 2 polynomial kernel and then the ones of a degree 3 polynomial kernel.

5.3.1 Poly Kernel, degree 2

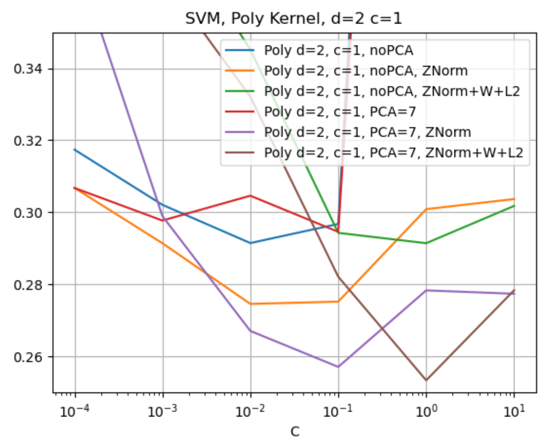
From the linear formulation of SVM we will pick $K = 1, \xi = 1$; we will then perform cross validation to find the best values of c to reduce costs as much as possible.

$$k(x_1, x_2) = (x_1^T x_2 + c)^d \tag{5.1}$$

Once picked the value of c, we will then continue by expoliting PCA and data preprocessing to see if the cost can be lowered even more.



As shown in the picture, best results are reached for $c = 1$. We will now use this set of values to apply PCA and data preprocessing. A zoomed graph will be reported to pick the best configuration.



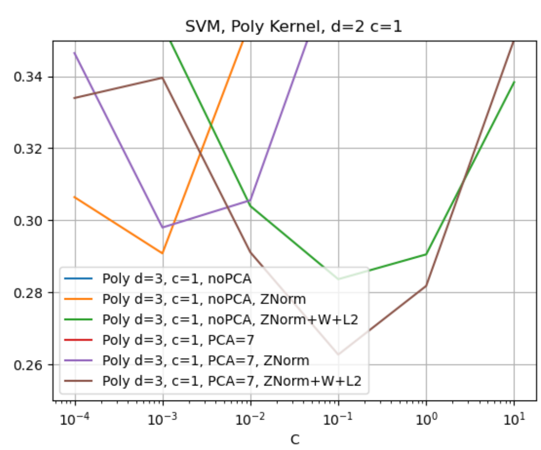
As we can see from the graph, both strategies help the task. In praticular our best performance so far is obtained with the following configuration: PCA = 7 and Z-Norm, Whitening and Length Normalization applied before processing.

C	PCA	Preoprocessing	minDCF($\tilde{\pi} = 0.09$)	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$
	7	NO	0.295	0.099	0.301
10^{-1}	7	Z-Norm	0.257	0.088	0.192
1	**7**	**Z-Norm + W + L2**	**0.253**	0.105	0.296

The classification seems to benefit from data preoprocessing for our applications. For the other two considered working points the results reached with it are actually worse.

5.3.2 Poly Kernel, degree 3

For degree 3 poly kernel we will follow the same strategy employed for the degree 2 one. Following a graph with all the considered configurations.



PCA dimentions	Preoprocessing	minDCF($\tilde{\pi} = 0.09$)
NOPCA	NO	0.560
NOPCA	Z-Norm	0.291
NOPCA	Z-Norm + W + L2	0.384
7	NO	0.392
7	Z-Norm	0.298
7	**Z-Norm + W + L2**	**0.263**

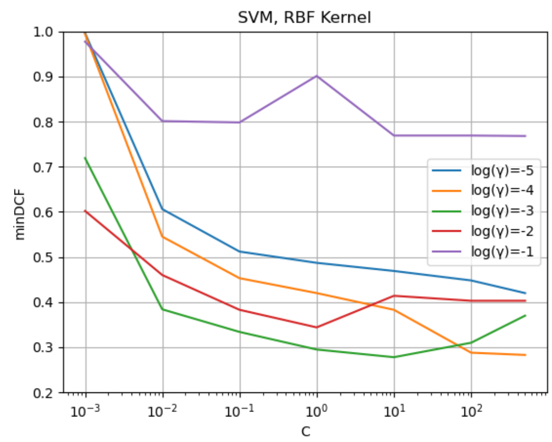
Once again we can see that best results are reached with PCA and data preprocessing together, but those are worse than the degree 2 Poly Kernel for our application parameters, so we will stick to that. (Results for other applications will not be shown)

5.3.3 RBF Kernel

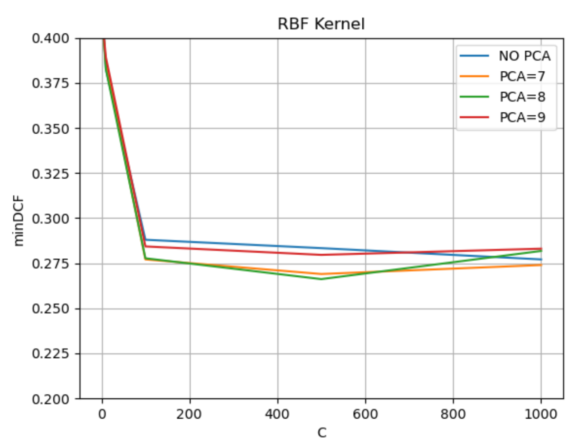
First we will pick the best best values for lambda.

$$k(x_1, x_2) = e^{-\gamma ||x_1 - x_2||^2} \tag{5.2}$$

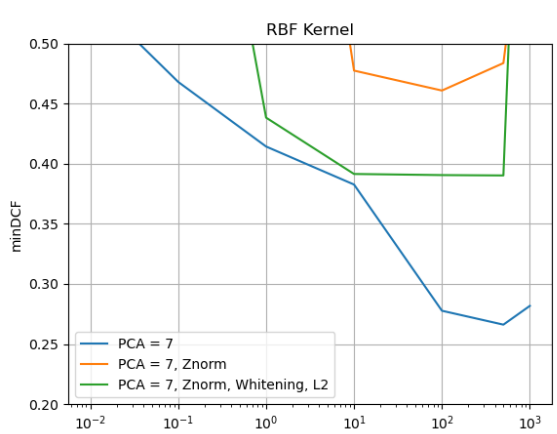
We will use cross validation to check which configuration brings us to best results.



The best results are reached for $\lambda = 10^{-3}$ and $\lambda = 10^{-4}$. From the reported plot we can see that the configuration with $\lambda = 10^{-4}$ might still not have reached a minimum; therefore, while applying PCA e preprocessing, we will increase the value of C, neglecting the small ones, to check if better results can be obtained.



The minimum value of DCF is 0.266 and is reached with $PCA = 8$ and $C = 500$. For this configuration we will apply data preprocessing and see if results improve even more.



Since data preoprocessing actually degrades our classification task we will stick to non preprocessed data. The reached cost is still higher than the one reached with Poly(2), therefore the model will be discarded.

5.4 Rebalancing

We can now try to embedd inside SVM scores prior information. As we did for the Quadratic Logistic Regression we will use different priors and check the performance for different applications. We will do this only for the best cofiguration obtained in this chapter: Poly(2) with PCA = 7, Z-Norm, Whitening, L2.

Prior	Kernel	PCA	minDCF($\tilde{\pi} = 0.09$	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$)
Fold	Poly	7	0.253	0.105	0.296
0.09	Poly	7	0.278	0.109	0.330
0.2	Poly	7	0.260	0.106	0.320
0.5	Poly	7	0.269	0.103	0.296
0.7	Poly	7	0.281	0.105	0.294

From the table we see that rebalancing the scores using different priors does not reduce the costs for our application parameters. For this reason we will stick to the non balanced verision of the SVM.

Chapter 6

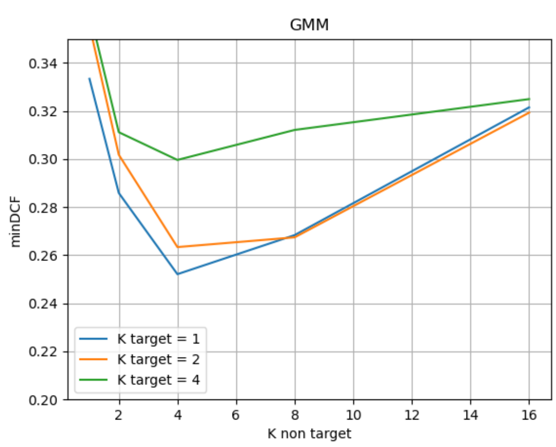
Gaussian Mixture Models

6.1 Introduction

In this chapter we will make use of Gaussian Mixture Models to classify data. For both the target and the non target class, we will consider Full Covariance, Diagonal and Tied models. Given the results obtained in chapter 2, we do not expect great results with D and T models.

6.2 Selection of components

We will firstly select the amount of gaussian components for each class. Following is a plot of the various configurations.



Our best configuration is: 1 component for target class and 4 components for non target class. We will now check if the usage of PCA is of any help.

PCA dimentions	minDCF($\tilde{\pi} = 0.09$)	minDCF($\tilde{\pi} = 0.5$)	minDCF($\tilde{\pi} = 0.9$)
NOPCA	0.252	0.078	0.152
9	0.257	0.081	0.140
8	0.264	0.079	0.159
7	**0.246**	0.079	0.159
6	0.251	0.077	0.151

PCA is indeed helping the classification, reaching the best results so far for all the models for our task. For the other working points, PCA does not seem to help in any way.

6.3 Diagonal or Tied

We will not apply diagonalization to non target class because, as we have seen from the histograms reported in chapter 1, features seem to be correlated for that class. What we will do is try the tied model for the Non Target class, and both Full Covariance and Diagonal for the Target class. We will only compute results using the best configuration obtained in the previous section.

Target	Non Target	minDCF($\tilde{\pi} = 0.09$)	minDCF($\tilde{\pi} = 0.5$)	minDCF($\tilde{\pi} = 0.9$)
FC	FC	0.246	0.079	0.159
FC	T	0.284	0.084	0.160
D	FC	0.605	0.168	0.252
D	T	0.603	0.169	0.256

None of the assumption is lowering our score so we will only consider the score obtained with both classes modelled with Full Covariance components.

Chapter 7

Calibration

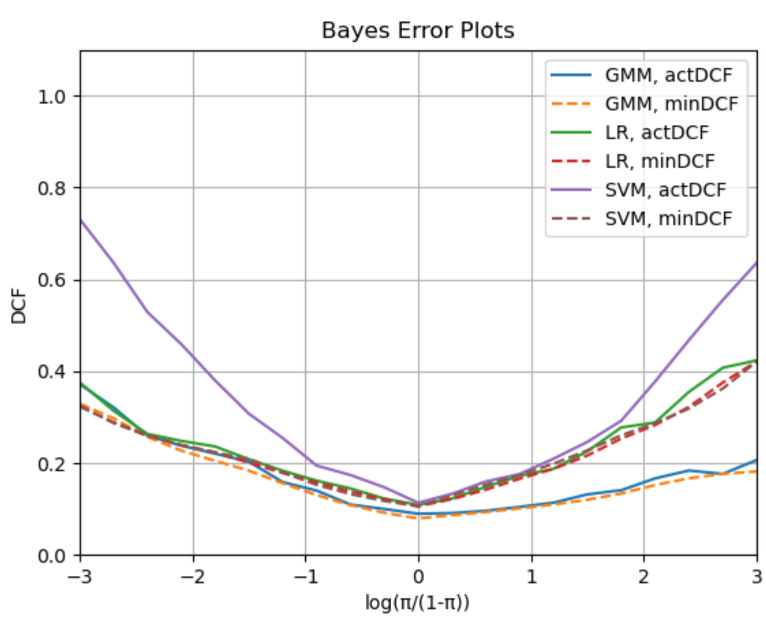
As of now, these are our best models (in the following tables, PRE stands for Z-Norm, Whitening, Length Normalization):

Model	minDCF($\tilde{\pi} = 0.09$)
QuadLogReg ($\lambda = 10^{-4}, PCA = 7$, PRE)	0.252
Poly($d = 2, c = 1, C = 1, PCA = 7$, PRE)	0.253
GMM($K_0 = 4, K_1 = 1$)	0.246

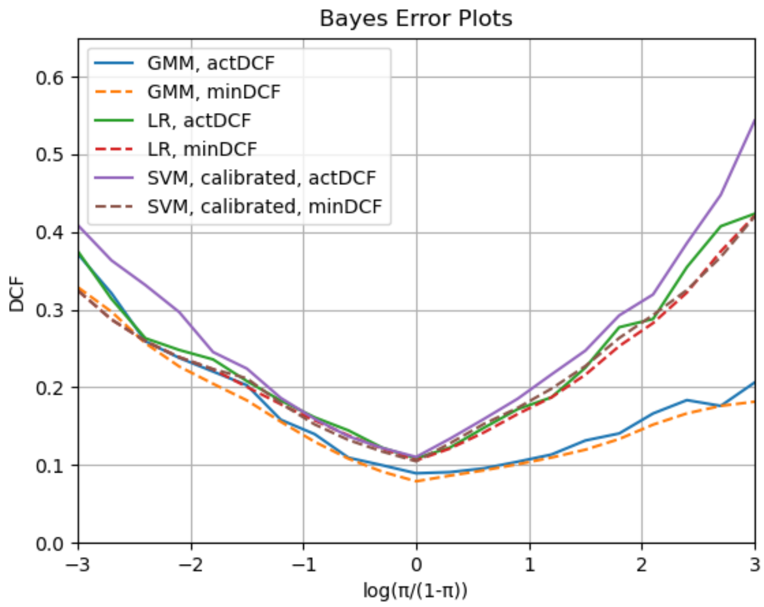
In this section we will calculate the actualDCF for our application and check if any of the models in the table above needs calibration. We will perfrom the checks only for our target. Performances of the classifiers for other applications will be checked on the Bayes Error Plots.

Model	minDCF($\tilde{\pi} = 0.09$)	actDCF($\tilde{\pi} = 0.09$)
QuadLogReg ($\lambda = 10^{-4}, PCA = 7$, PRE)	0.252	0.262
Poly($d = 2, c = 1, C = 1, PCA = 7$, PRE)	0.253	0.503
GMM($K_0 = 4, K_1 = 1$)	0.246	0.253

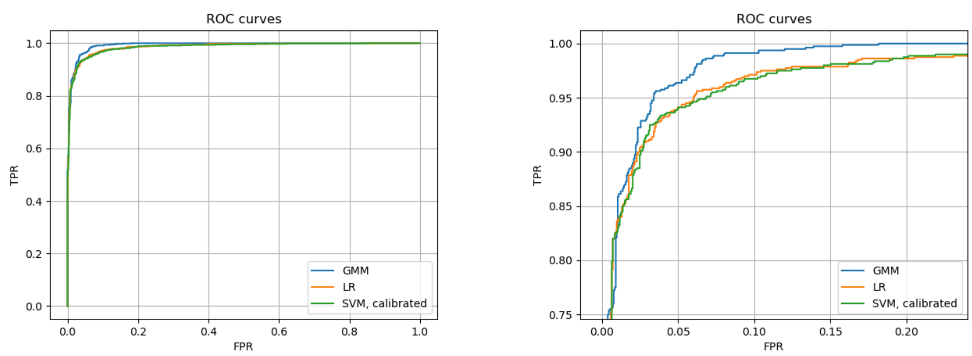
Following are the Bayes Error Plots for the chosen models.



From the plots we understand that both the Quadratic Logistic Regression model and the Gaussian Mixture model are well calibrated for a wide range of applications (including ours). On the other hand, the Poly Kernel SVM model is not well clalibrated for our applicationl; it is calibrated for application with $\tilde{\pi} \in [0.5, 0.8]$. Therefore, we will perform calibration for it.



After the procedure we can see that all the models are calibrated for a wide range of application and we can finally move to evaluation. Following are the ROC curves for the three models.



(a) ROC (b) ROC, Zoomed

Our best model, suggested by the curves is the GMM model.

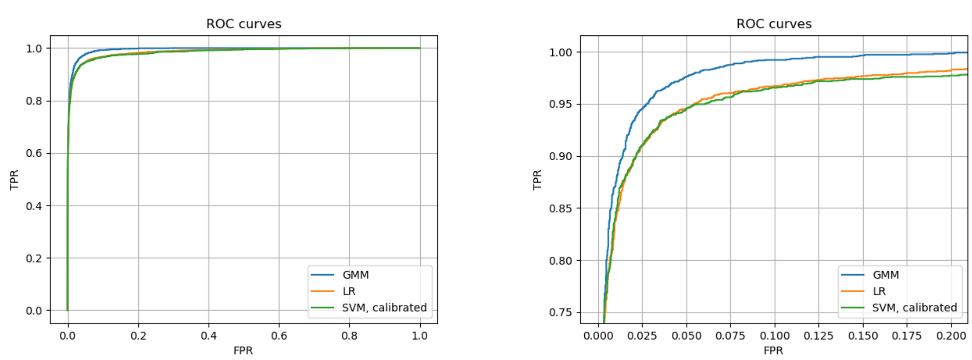
Chapter 8

Evaluation

In this last chapter we will perform evaluation of our best models. In the following table the $minDCF$ and $actDCF$ costs, evaluated both on the training set and on the test set, are reported. All the cost functions are evaluated using our application effective prior.

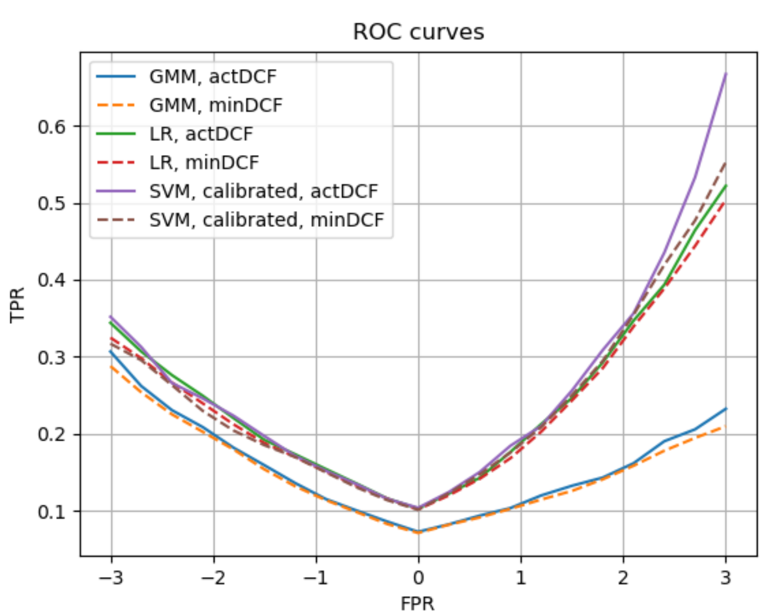
Model	minDCF(train)	actDCF(train)	minDCF(test)	actDCF(test)
QuadLogReg	0.252	0.262	0.256	0.271
Poly(With Calibration)	0.253	0.318	0.252	0.261
GMM	0.246	0.253	0.217	0.239

All three models reach results that are similare, both on the training and the test set. The Gaussian Mixture model actually reaches costs which are lower than the ones obtained on the training set. Overall the results are consistent and the performances are similar.



(a) ROC (b) ROC, Zoomed

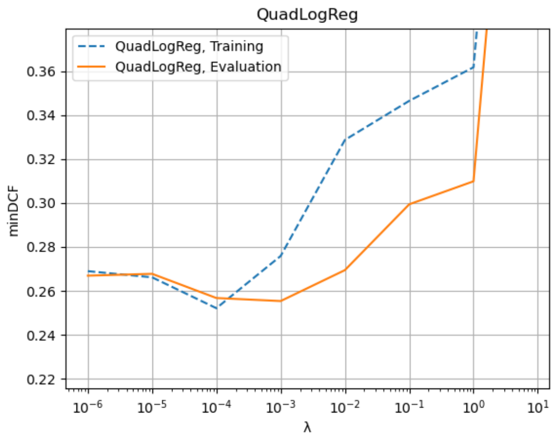
From the ROC curve, once again we see that the best model is GMM model.



In the following sections we will analyize our choices for the models hyperparameters, to understand if they were optimal or better results could have been reached.

8.1 Quadratic Logistic Regression

We will frist analyze if the choices made for the value of λ were correct.



Our chosen value is $\lambda = 10^{-4}$, but the actual minimum for the evaluation set is reached for $\lambda = 10^{-3}$. With this value we would have reached $minDCF = 0.255$, very close to our actual minDCF reached on the evaluation set of 0.257. The choice of λ was effective.

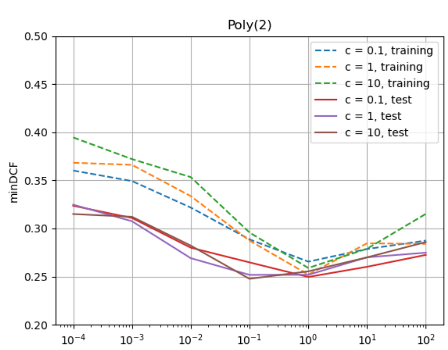
Lastly we will check if PCA has helped or worsen our task.

PCA dimensions	minDCF(train)	minDCF(test)
No PCA	0.289	0.262
9	0.281	0.269
8	0.278	0.259
7	0.252	**0.256**
6	0.255	0.249

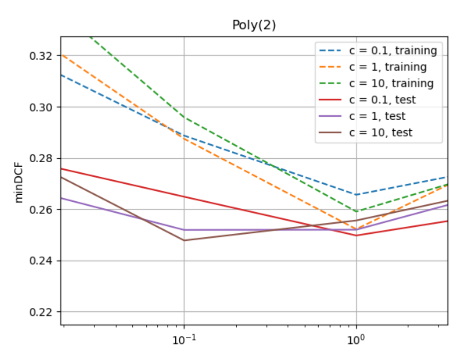
PCA with 6 dimensions would have gotten ourself better results than the ones obtained with 7 dimensions; as for λ the difference in cost is still very small.

8.2 Poly Kernel SVM

For the Poly(2) Kernel SVM we will firstly analize the choices made for the parameter c inside the kernel formulation and for the hyperparameter C.



(c) minDCF, Poly(2)



(d) minDCF, Poly(2), zoomed

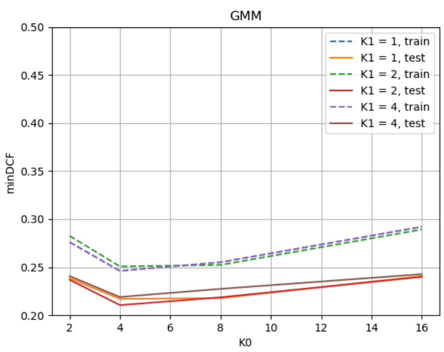
The minimum cost would have been reached with $c = 10$ and $C = 10^{-1}$ (minDCF $=$ 0.247) instead of $c = 1$ and $C = 1$ (minDCF $= 0.252$). The difference is once again very small. We will procede by analyzing the choices made for the PCA dimensions.

PCA dimensions	minDCF(train)	minDCF(test)
No PCA	0.293	0.257
9	0.280	0.260
8	0.284	0.256
7	0.252	**0.252**
6	0.265	0.250

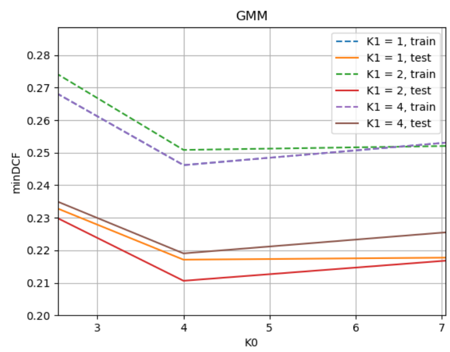
As for all previous case, the reached cost is almost equal to the obtained one.

8.3 Gaussian Mixture Model

First we will check if our choices for $K0$ and $K1$ were correct.



(e) minDCF, GMM



(f) minDCF, GMM, Zoomed

On the evaluation set the minimum cost is reached for $K1 = 2$ instead of $K1 = 1$. As we did for the two previous models, the next step will be checking if the picked PCA dimension is correct.

PCA dimensions	minDCF(train)	minDCF(test)
No PCA	0.252	0.220
9	0.257	0.219
8	0.264	0.218
7	0.246	**0.217**
6	0.251	0.221

$PCA = 7$ seems indeed correct. We will lastly check if Tied or Diagonal assumptions would have helped our task.

Target	Non Target	minDCF(train)	minDCF(test)
FC(1)	T(4)	0.284	0.230
FC(1)	T(8)	0.252	0.222
FC(1)	T(16)	0.246	0.223
D(2)	T(4)	0.610	0.475
D(2)	T(8)	0.606	0.473
D(2)	T(16)	0.604	0.472
D(4)	T(4)	0.591	0.459
D(4)	T(8)	0.585	0.452
D(4)	T(16)	0.589	0.453

None of the reported condfiguration would have decreased the cost for the classification.

8.4 Conclusions

Our overall approach was successful. Our best model is the Gaussian Mixture Model with $K0 = 4, K1 = 1$. The results achieved on the Test Set are the following:

1. DCF = 0.239
2. Accuracy = 94.77 %
3. Error Rate = 5.23 %

We will report the final confusion matrix.

	Pred 0	Pred 1
Class 0	5255	354
Class 1	49	2046

As we can see from the table, the amount of False Positive is small compared to the amount of False Negative, in line with our application mismatched costs.