Language Detection 2023

Machine Learning and Pattern Recognition

Guido Genco
s285703@studenti.polito.it

Francesco Renda
s292500@studenti.polito.it

Politecnico di Torino

July 2023

Summary

1 Introduction

The goal of this project is the development of a classifier to detect whether an utterance is spoken in a given target language or not. The target language is Italian. The utterances are represented by means of language embeddings, the dataset consists of synthetic data, and embeddings have significantly lower dimension than in real use-cases. The embeddings are **6-dimensional**, continuous-valued vectors, belonging to either the target language or the non-target language class. The embedding components do not have a physical interpretation.

The training set consists of 400 embeddings for target class (Italian) and 1971 embeddings for non target class (not Italian), for a total of 2371 embeddings. The test set consists of 800 embeddings for target class (Italian) and 3603 embeddings for non-target class (not Italian), for a total of 4403 embeddings.

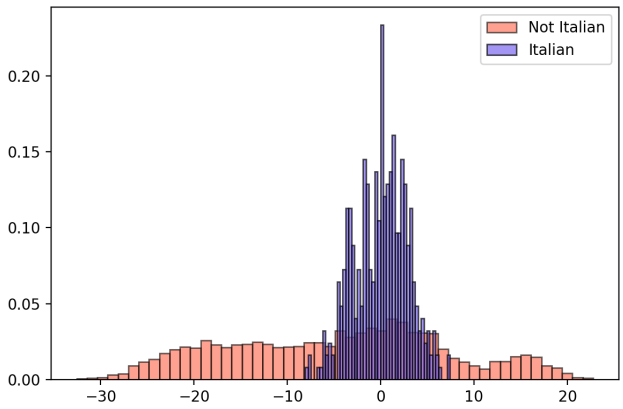
Two possible target applications:

- $(\pi = 0.1, Cfn = 1, Cfp = 1)$
- $(\pi = 0.5, Cfn = 1, Cfp = 1)$

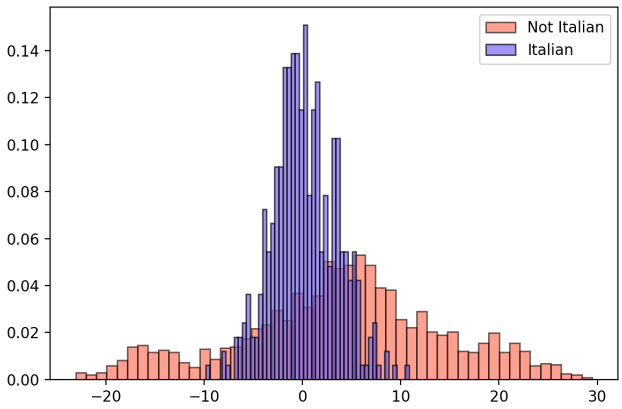
All results reported in the section 3 have been obtained performing a K-fold cross validation with K=5. The primary metric that will be used to compare models performances is C_{prim} which is the average minDCF of the two working points listed above. Different models will be evaluated and compared in order to choose the most suitable for the task

2 Features analysis

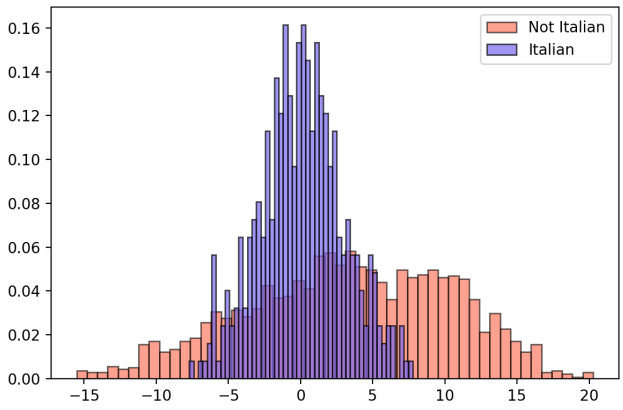
We start analyzing the features of our data. As already mentioned above, we have 6-dimensional embeddings. An histograms for each feature is plotted below:



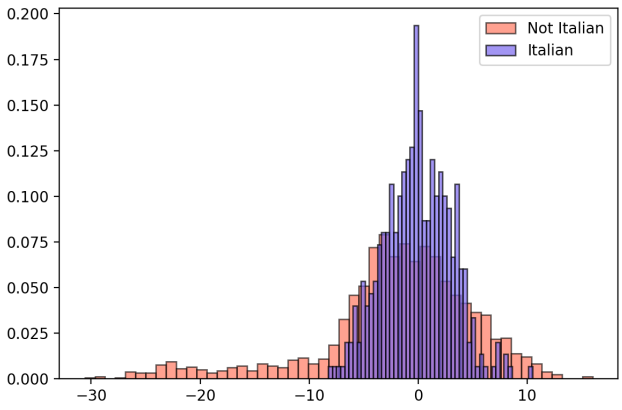
RAW feature 1



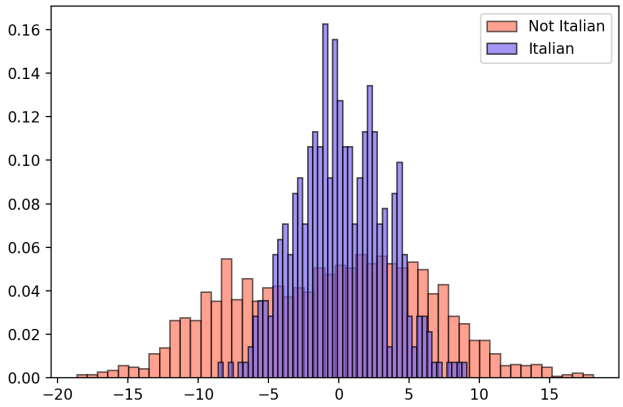
RAW feature 2



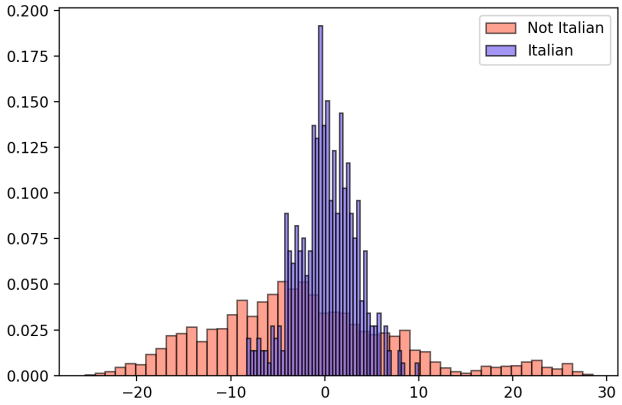
RAW feature 3



RAW feature 4



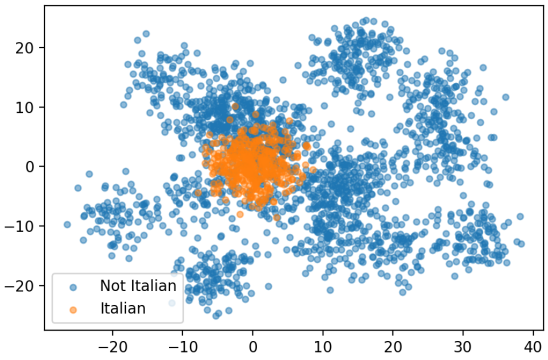
RAW feature 5



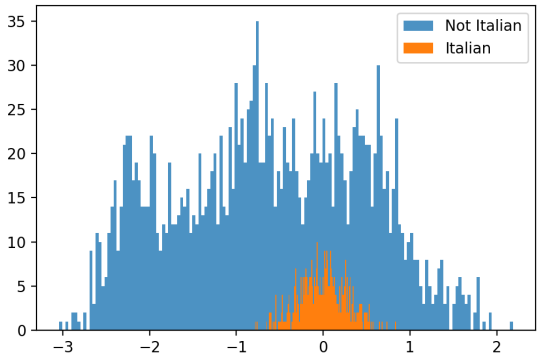
RAW feature 6

it's evident that the target class features can be approximated with a Gaussian distribution, on the other hand for the non-target class features the same kind of distribution is not a good approximation.

In order to have a global view of our training data, PCA (with 2 dimensions) and LDA have been exploited. The corresponding plots are reported below:



PCA applied on training set



LDA direction

LDA shows that features are not linearly separable and for this reason we expect to obtain poor performances from linear models like for example Linear Logistic Regression. Given the distribution of the features shown in the PCA scatter plots we can conclude that target and non-target features overlaps, moreover the non-target class is characterized by multiple clusters while the target class by a single one. We expect non-linear models to be significantly better for the task.

3 Validation phase

3.1 Gaussian Classifiers

The first classifiers family that we analyze is Multivariate Gaussian, specifically we compare performances of 3 different models:

- Multivariate Gaussian Classifier (standard)
- Naive Bayes Gaussian Classifier
- MVG with Tied Covariances

3.1.1 Expectations

We expect to see better results with the MVG standard version because, as mentioned before, our target class features can be approximated with a Gaussian distribution and also because we have seen that they are not linearly separable, indeed the MVG standard makes possible to obtain decision boundaries more suitable for our dataset, we also expect to obtain the worst performance with the Tied version for the same reason.

3.1.2 Results

Results obtained with and without PCA are reported in the table below:

Version	PCA	minDCF($\pi = 0.1$)	minDCF($\pi = 0.5$)	C_{prim}
MVG	-	0.483	0.124	0.303
	5	0.472	0.120	0.296
	4	0.544	0.127	0.336
	3	0.525	0.127	0.326
	2	0.864	0.191	0.528
Naive Bayes	-	0.537	0.139	0.338
	6	0.478	0.126	0.302
	5	0.47	0.127	0.298
	4	0.534	0.131	0.332
	3	0.53	0.129	0.329
	2	0.866	0.192	0.529
MVG Tied	-	1.0	0.515	0.758
	5	1.0	0.517	0.759
	4	1.0	0.478	0.739
	3	1.0	0.477	0.739
	2	1.0	0.498	0.749

As expected the MVG performs better than the other two models and the worse performances are the ones obtained with the Tied MVG. It's noticeable that PCA improves results but only keeping 5 dimension, less than 5 means worse performances. For Naive Bayes also results for PCA 6 are reported because performing PCA and keeping the same number of dimensions as the one of the original features diagonalizes the covariance matrix of the dataset, and this seems to improve results with the Naive Bayes while with the other ones results with PCA 6 were exactly the same obtained without PCA so we omit them.
Results obtained performing Z-Norm as pre-processing with best configurations for both MVG standard version and Naive Bayes are reported in order to see if this can bring improvements in terms of performances, we omit MVG Tied because its results with raw features were not good:

Version	PCA	minDCF($\pi = 0.1$)	minDCF($\pi = 0.5$)	C_{prim}
MVG	5	0.557	0.139	0.348
Naive Bayes	5	0.522	0.135	0.329

Z - Norm

It's clear that Z-Norm is not useful. Finally, we can conclude that the best performances have been obtained with **MVG applying PCA 5 without Z-Norm**

3.2 Linear and Quadratic Logistic Regression

In this section results obtained with **Prior-weighted** Logistic Regression both **Linear** and **Quadratic** version will be discussed. The main difference between those two versions is the feature expansion process that characterizes the Quadratic one:

$$\phi(x) = \begin{bmatrix} vec(xx^T) \\ x \end{bmatrix}$$

$\phi(x)$ is performed on each feature vector x in order to build an expanded feature space where classes are linearly separable, computing linear separation rules for the expanded feature space means computing quadratic separation surfaces in the original feature space. In order to achieve this we need to train our Logistic Regression model using $\phi(x)$ instead of x.

A grid search style was used in order to define which is the best parameters configuration for this two models considering different values of Prior π, regularization parameter λ and PCA.
Training has been performed with 5 different values of Prior π:

- 0.1
- 0.5
- Dataset Prior
- Fold Prior
- 0.2

the latter is a value somewhere between the two working points that we consider.

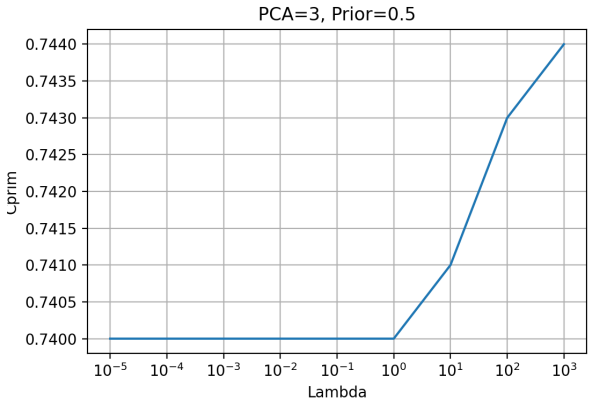
3.2.1 Expectations

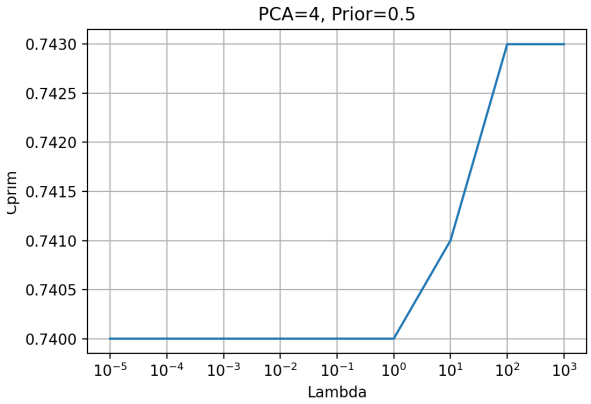
Considering that Linear Logistic Regression is a linear model we don't expect to see good results, quite the contrary. Quadratic Logistic Regression, on the other hand, we believe will produce quite good results because, as mentioned previously, this model is able to develop non-linear separation rules.

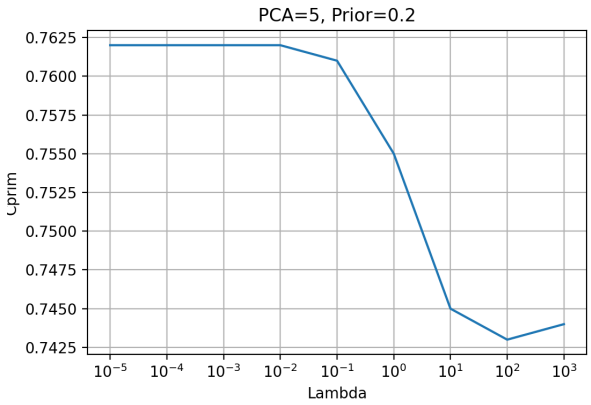
3.2.2 Results

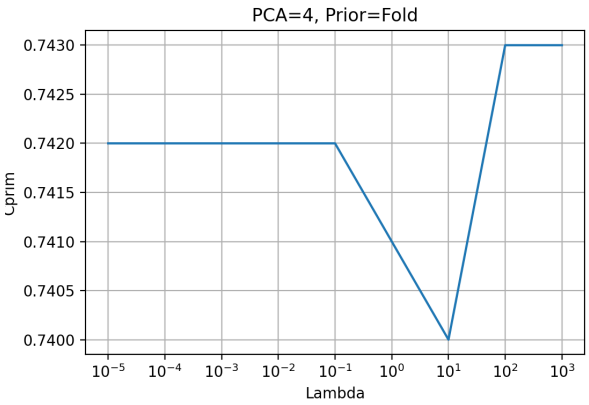
The grid search approach produced a large amount of results, for this reason in this section the complete results table has been omitted but instead plots have been reported, because they allow to have a complete picture of performances computed with this two versions of Logistic Regression. Each plot shows different values of C_{prim} obtained for different values of λ given a Prior π and a PCA applied.

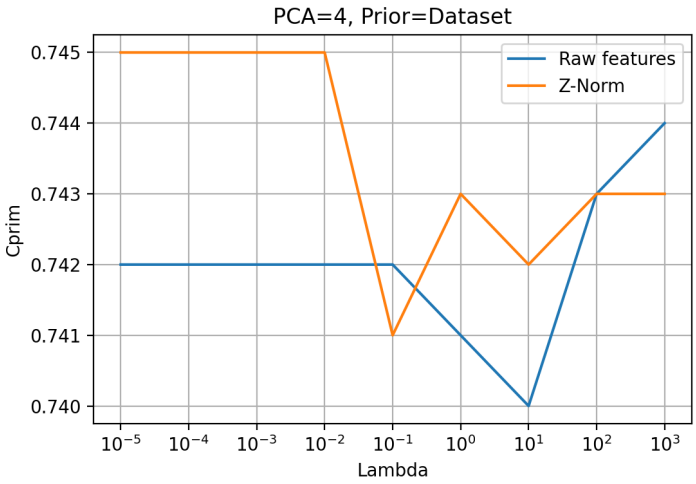
Performances computed with Linear Logistic Regression are now reported





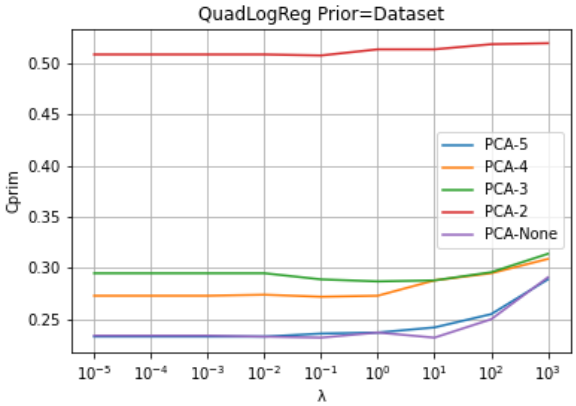


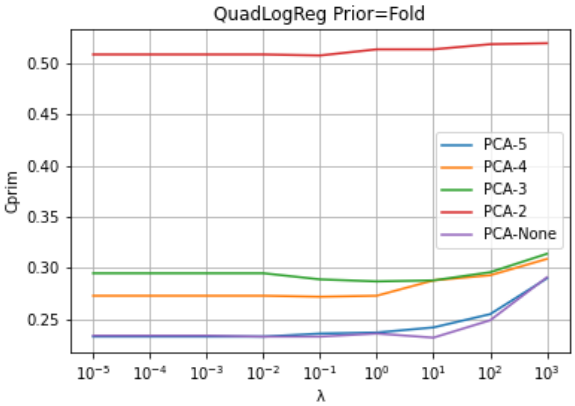


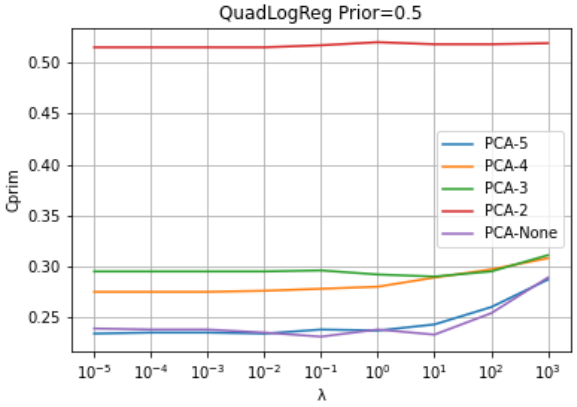


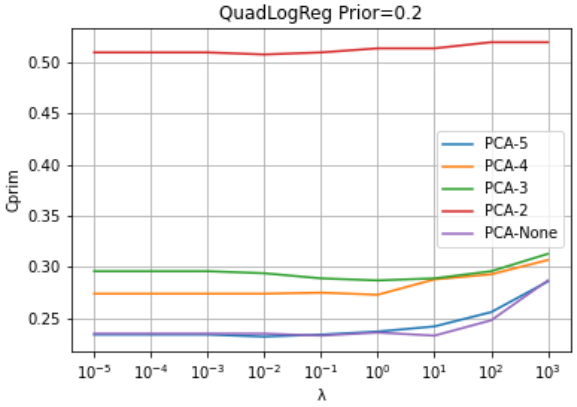
The best C_{prim} value obtained was 0.74 and only plots related to configurations that achieve this value have been reported above. As expected the Linear Logistic Regression is not able to perform well on our data because of its linearity. Since results are bad, only in the latter plot a comparison between validation performed with and without Z-Norm pre-processing is reported and, as can be seen from the graph, this technique does not improve results at all.

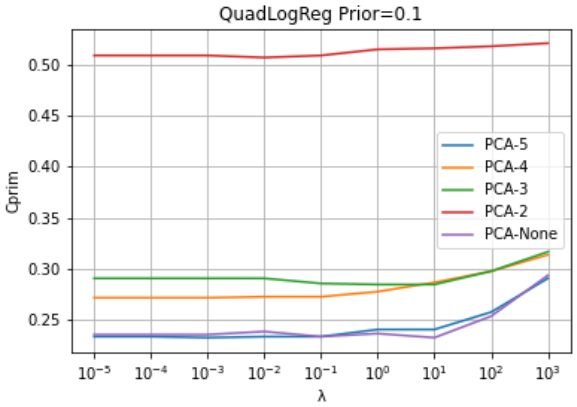
Moving to Quadratic Logistic Regression, obtained results are reported below:











Results computed with Quadratic Logistic Regression, as expected, are better than the ones obtained with the Linear version, this we think happened because of the quadratic decision boundaries that characterize this version of the model, indeed they are more suitable for our data. **We applied PCA before feature expansion** and PCA 5 seems to be effective: we get results close to (but still worse in general) the ones obtained without dimensionality reduction. **The best result has been obtained with the following configuration**:

Prior π	PCA	λ	Z-Norm	minDCF($\pi = 0.1$)	minDCF($\pi = 0.5$)	C_{prim}
0.5	No	0.1	No	0.381	0.081	0.231
			Yes	0.463	0.137	0.300

Best QuadLogReg configuration

We also tested on this configuration the effectiveness of Z-Norm but reported results show that even in this case Z-Norm does not bring any benefits.

Finally we can conclude at the end of this section that results obtained with Quadratic Logistic Regression are definitely good if compared with the Linear version ones, actually **Quadratic Logistic Regression performed better than any other considered model so far**, obtaining relatively close but anyway better results than the ones computed with MVG

3.3 SVM

In this section results obtained with SVM models will be discussed. Specifically, we used three SVM versions:

- Linear
- Polynomial kernel of degree d with kernel: $k(x_1, x_2) = ({x_1}^T x_2 + c)^d$
- Radial Basis Function kernel (RBF): $k(x_1, x_2) = e^{-\gamma \|x_1 - x_2\|^2}$

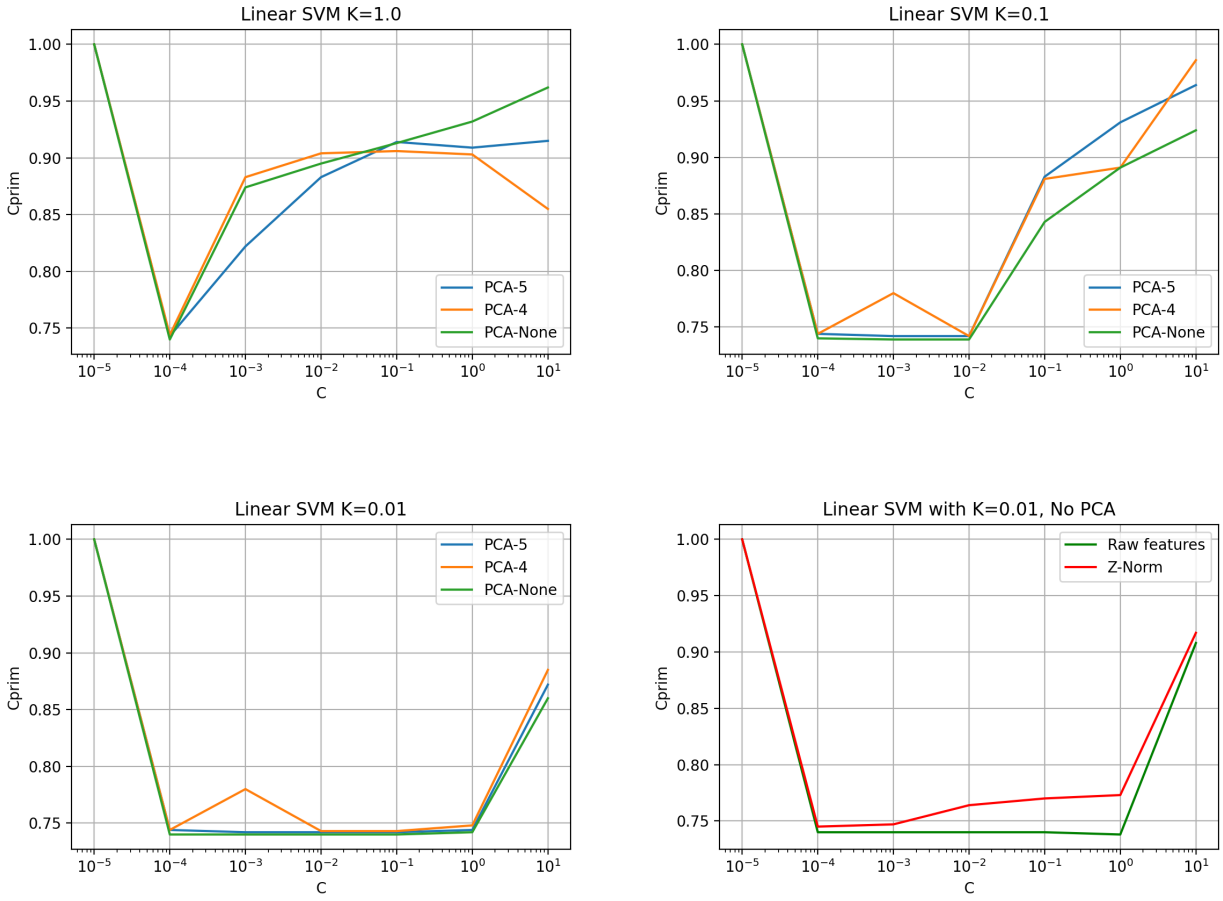
Also with these models a grid search approach has been exploited, for this reason even in this section it was preferred to report results in the form of graphs rather than in tabular form, even tho best result are reported also in tabular form.

3.3.1 Expectations

For what concerns Linear SVM we expect to see results similar to the ones computed with Linear Logistic Regression, on the other hand we expect to see better results with the two before introduced kernel versions because the kernel function in SVMs allow for non-linear classification through an implicit expansion of the features in a higher-dimensional space.

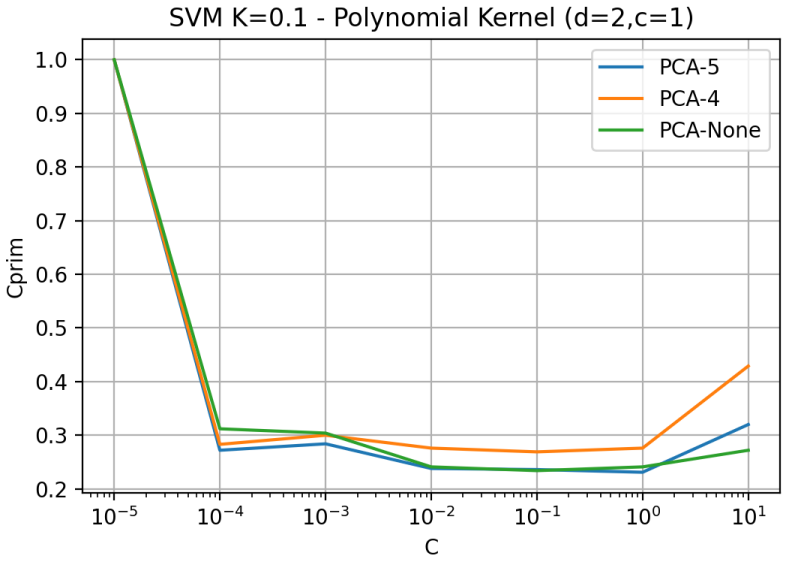
3.3.2 Results

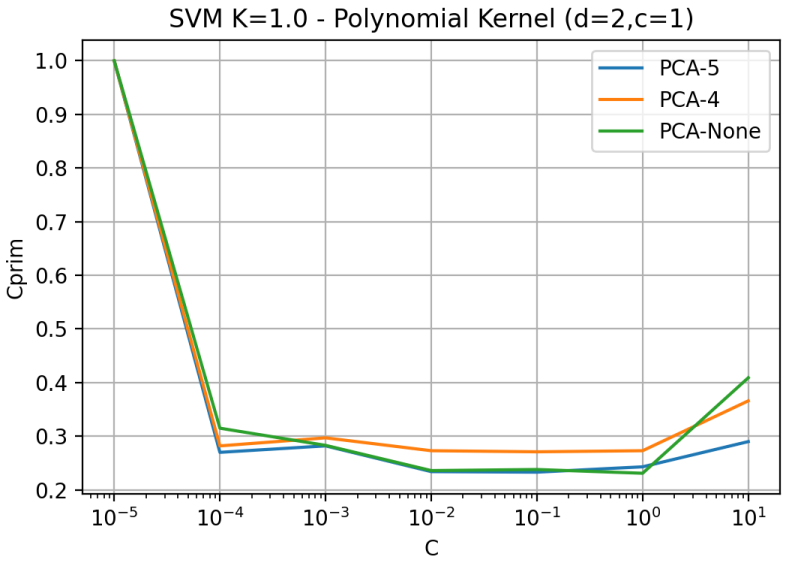
Firstly, **Linear SVM** results are reported:

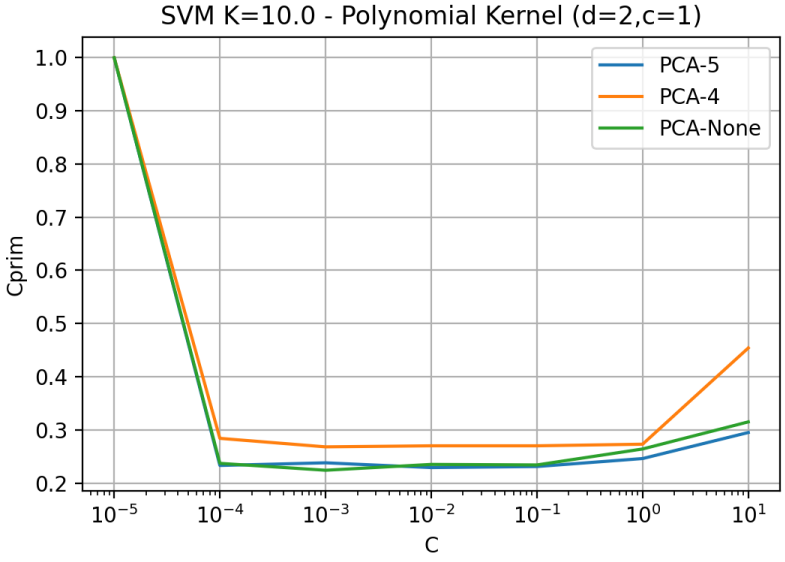


As expected results are really close to Linear Logistic Regression results, they are not at all exiting and we believe that the cause is the linearity of the model. Neither in this case Z-Norm seems to improve results as shown in the latter plot. PCA seems to achieve results similar to the ones obtained without dimensionality reduction both with 5 and 4 dimensions in general.

Polynomial kernel SVM results are now discussed. Plots related to configurations that achieve best results are reported below:



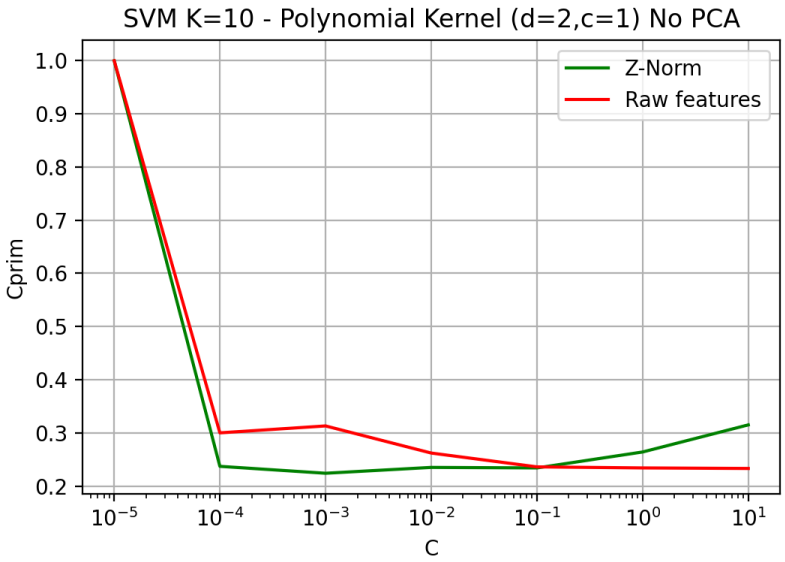




K	C	Kernel	PCA	minDCF($\pi = 0.1$)	minDCF($\pi = 0.5$)	C_{prim}
0.1	1.0	Poly(d=2,c=1)	5	0.367	0.094	0.231
1.0	1.0	Poly(d=2,c=1)	-	0.374	0.089	0.231
10.0	0.1	Poly(d=2,c=1)	5	0.369	0.092	0.231
10.0	0.01	Poly(d=2,c=1)	5	0.367	0.091	0.229
10.0	0.001	Poly(d=2,c=1)	-	0.361	0.087	0.224

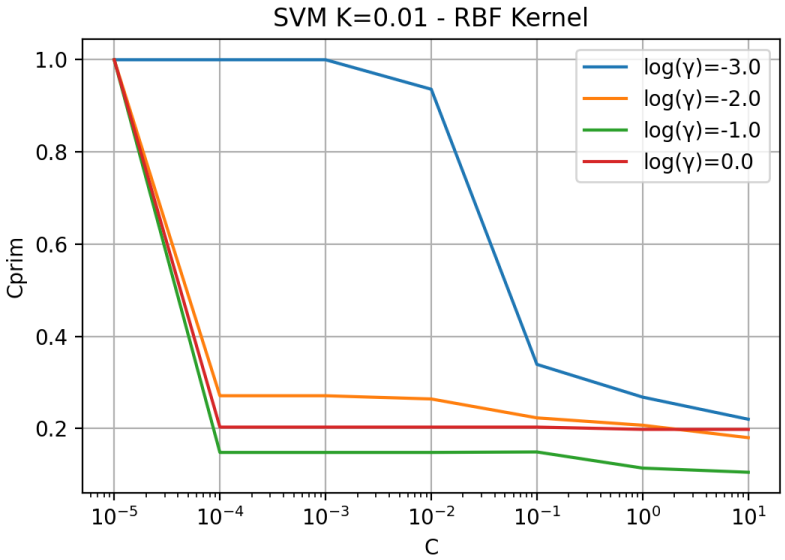
SVM Polynomial best results table

The effects of Z-Norm on the best parameters configuration for SVM Polynomial are reported below:

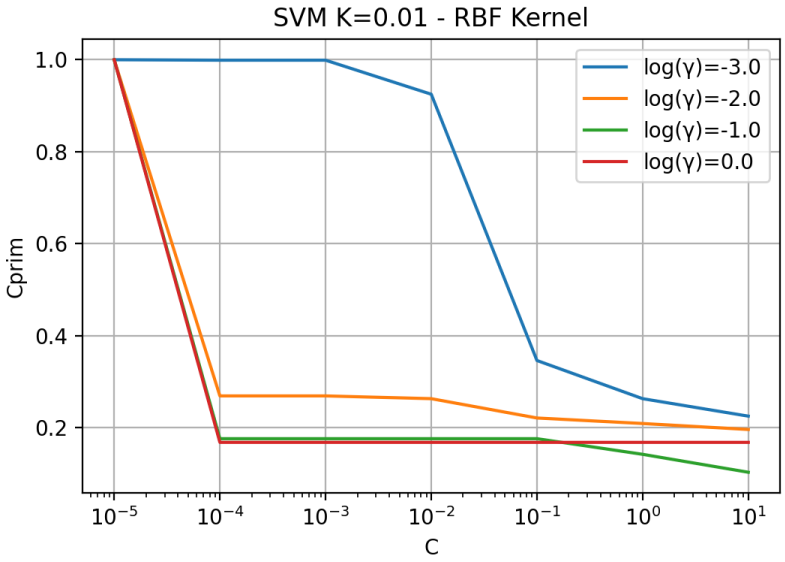


In this case Z-Norm seems to achieve similar results to the ones computed without Z-Norm but does not improve anyway performances: the best C_{prim} value computed with Z-Norm is 0.233 with C = 10, worse than the minimum obtained without pre-processing.

Finally, results computed with **RBF SVM** are reported:



RBF SVM Raw features



RBF SVM with PCA 5

Since a lot of plots have been generated only those related with the best results shown in the table below have been reported.

K	C	Kernel	PCA	minDCF($\pi = 0.1$)	minDCF($\pi = 0.5$)	C_{prim}
0.01	10.0	RBF($\gamma = 0.1$)	5	0.149	0.058	0.103
0.01	10.0	RBF($\gamma = 0.1$)	-	0.156	0.054	0.105
1.0	10.0	RBF($\gamma = 0.1$)	5	0.156	0.055	0.105
10.0	10.0	RBF($\gamma = 0.1$)	5	0.156	0.055	0.105
0.1	10.0	RBF($\gamma = 0.1$)	5	0.158	0.054	0.106

SVM RBF best results table

Also in this case Z-Norm has been tested with the best configuration of RBF SVM but in this case seems that results are negatively influenced by this pre-processing technique:

K	C	Kernel	PCA	minDCF($\pi = 0.1$)	minDCF($\pi = 0.5$)	C_{prim}
0.01	10.0	RBF($\gamma = 0.1$)	5	0.985	0.981	0.983

Z-Norm SVM RBF

To summarize, results computed with SVM models show that the Linear version, as expected, is not suitable for our data and its performances are close to other linear models ones presented before. On the other hand, the kernel functions make possible to achieve great performances on our data, specifically **RBF SVM outperforms not only the Polynomial version but also any other model considered to this point**. Also PCA seems to be effective in this case with a dimensionality of 5.

3.4 Gaussian Mixture Models

In this section Gaussian Mixture Models (GMM) results will be presented and discussed. We will use Full GMM, Tied GMM and also Diagonal GMM.

3.4.1 Expectations

Since our target class data, as said in section 2, can be approximated with a Gaussian distribution we expect to see good result. Considering that is possible to choose different GMM number of components for different classes, different combinations have been tested. We expect that a small number of components for target class and a large number of components for non-target class should provide good performances. Tied GMM with more components we think could be a good solution for non-target class because data seems distributed according several clusters with similar distribution.

3.4.2 Results

For each tested combination of:

- number of GMM components for target and non-target class
- type of GMM (Full, Tied and Diagonal)

obtained results are reported in the tables below, specifically since each combination has been tested with and without PCA with different numbers of dimensions, only the best result for each combination is reported:

Target	Non-target	PCA	minDCF($\pi = 0.1$)	minDCF($\pi = 0.5$)	best C_{prim}
1-FC	8-FC	5	0.375	0.086	0.231
1-FC	16-FC	5	0.365	0.084	0.224
1-FC	32-FC	5	0.336	0.08	0.208
1-FC	64-FC	-	0.358	0.088	0.223
2-FC	8-FC	-	0.376	0.076	0.226
2-FC	16-FC	-	0.367	0.078	0.223
2-FC	32-FC	-	0.362	0.077	0.22
2-FC	64-FC	-	0.348	0.076	0.212

Full GMM results

Target	Non-target	PCA	minDCF($\pi = 0.1$)	minDCF($\pi = 0.5$)	best C_{prim}
1-D	8-D	-	0.407	0.092	0.249
1-D	16-D	-	0.37	0.084	0.227
1-D	32-D	-	0.332	0.078	0.205
1-D	64-D	-	0.324	0.072	0.198
2-D	8-D	-	0.411	0.092	0.251
2-D	16-D	-	0.367	0.08	0.223
2-D	32-D	-	0.331	0.078	0.205
2-D	64-D	-	0.324	0.072	0.198

Diagonal GMM results

Target	Non-target	PCA	minDCF($\pi = 0.1$)	minDCF($\pi = 0.5$)	best C_{prim}
1-T	8-T	-	0.402	0.096	0.249
1-T	16-T	5	0.367	0.09	0.228
1-T	32-T	5	0.317	0.09	0.203
1-T	64-T	-	0.301	0.076	0.189
2-T	8-T	-	0.402	0.096	0.249
2-T	16-T	5	0.367	0.09	0.228
2-T	32-T	5	0.317	0.09	0.203
2-T	64-T	-	0.301	0.076	0.189

Tied GMM results

For combination of different kind of GMM, only best results are reported in the next table:

Target	Non-target	PCA	minDCF($\pi = 0.1$)	minDCF($\pi = 0.5$)	best C_{prim}
1-F	64-D	-	0.324	0.072	0.198
1-F	64-T	-	0.301	0.076	0.189
1-T	32-F	5	0.336	0.08	0.208
1-T	64-D	-	0.324	0.072	0.198
1-D	32-F	5	0.336	0.08	0.208
1-D	64-T	-	0.301	0.076	0.189

Mixed GMM best results

Results confirmed our expectation, models with 1 component for target class and several components for non-target class obtained the best results. Also our idea about different clusters with similar distribution seems to be correct because Tied GMM for non-target class seems the most suitable choice. The best configuration is 1-Full component GMM for target class and 64-Tied components GMM for non target-class, without PCA. We test effects of Z-Norm on this configuration and results show that in this case this feature pre-processing technique is effective:

Target	Non-target	PCA	minDCF($\pi = 0.1$)	minDCF($\pi = 0.5$)	best C_{prim}
1-F	64-T	-	0.296	0.074	0.185

GMM best configuration with Z-Norm

Finally, we can conclude that **the best configuration for what concerns Gaussian Mixture Models is the one reported in the latter table**, results are good but not better than the RBF SVM computed results.

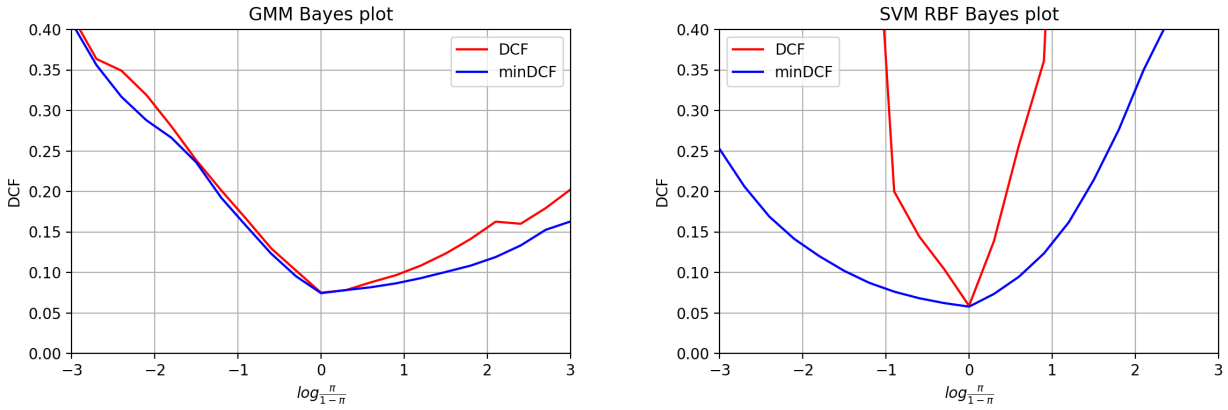
3.5 Comparing models

Considering results computed during the validation phase, our best models are basically two, each one is reported in the table below with its relative best configuration, minimum and actual DCF computed on validation set.

Model	**minDCF ($\pi = 0.1$)**	**minDCF ($\pi = 0.5$)**	**min C_{prim}**	**actDCF ($\pi = 0.1$)**	**actDCF ($\pi = 0.5$)**	C_{prim}
SVM RBF($\gamma = 0.1$) $K = 0.01$, $C = 10$, PCA = 5	0.149	0.058	0.103	0.985	0.059	0.522
GMM 1-F 64-T, Z-Norm	0.296	0.074	0.185	0.332	0.075	0.204

Models with best performances in validation phase

In general, our expectations have been satisfied because the models that better performed are not linear, specifically RBF SVM introduced non-linear decisions boundaries which are more suitable for our data, while GMM makes possible to model differently target and non-target class, which indeed have different distributions and different numbers of clusters. In order to compare these models and have a better idea of each model performances Bayes error plots can be exploited:

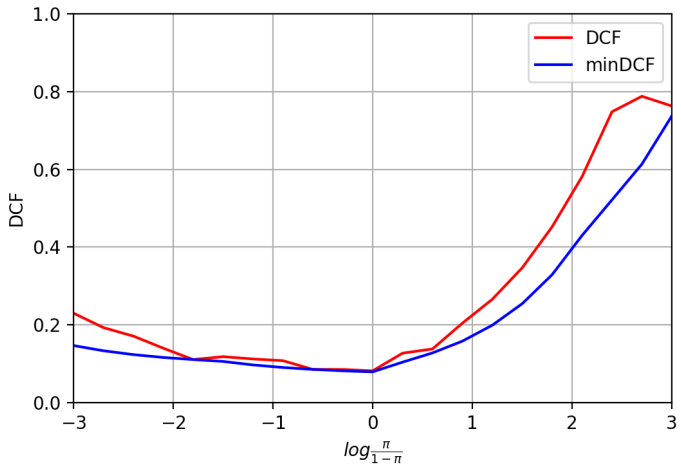


It's evident that for what concerns GMM, scores are already well calibrated since minDCF and DCF values are really close to each other, on contrary SVM scores are totally uncalibrated, specifically seems to achieve a good actual DCF only for a specific working point. SVM scores had to be calibrated and to do that an un-regularized Linear Log-Reg has been used: original validation scores computed after K-Fold training have been reshuffled, then a single split approach has been performed in order to use 80% of those scores to train the Linear Log-Reg and the remaining 20% as validation set for the calibration phase. Calibration has been performed with 3 different target priors: 0.1, 0.5 and 0.2. Results are reported below:

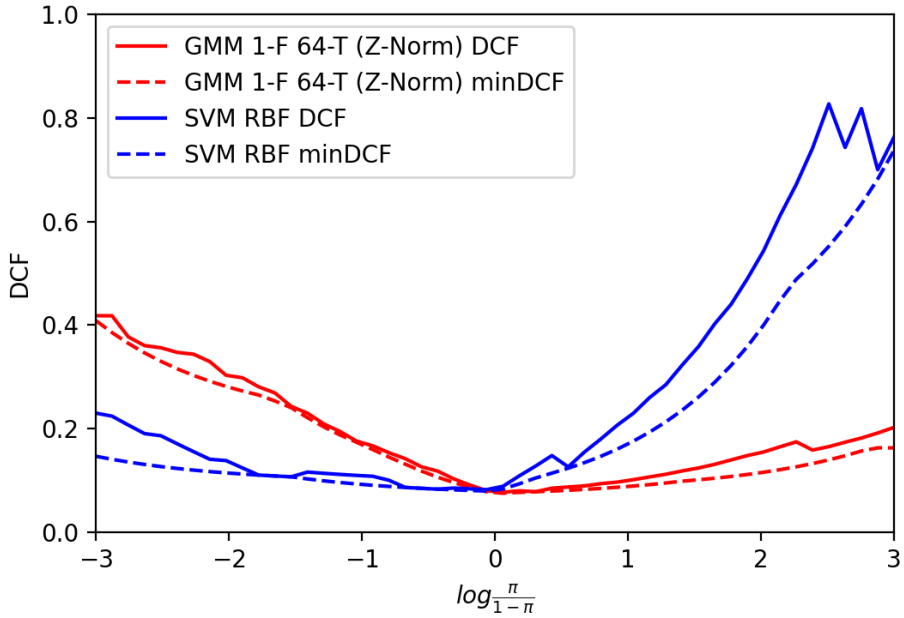
Target prior	minDCF ($\pi = 0.1$)	minDCF ($\pi = 0.5$)	min C_{prim}	actDCF ($\pi = 0.1$)	actDCF ($\pi = 0.5$)	C_{prim}
0.1	0.118	0.079	0.099	0.154	0.087	0.12
0.2	0.118	0.079	0.099	0.154	0.082	0.118
0.5	0.118	0.079	0.099	0.154	0.082	0.118

SVM RBF after score calibration

For target prior 0.5 and 0.2 results with score calibration are essentially the same, in general score calibration results don't seem to be influenced as much by the target prior. Since we are considering two working points we choose 0.2 as target prior for score calibration. The Bayes plot of SVM RBF after score calibration with prior 0.2 is reported below:



Finally, the Bayes plot containing GMM and score calibrated SVM RBF is reported:



GMM and RBF SVM are the models that we choose since they performed better than the others during validation phase, specifically the best model seems to be RBF SVM and for this reason we select it as final model. In the next chapter these models will be tested on evaluation set: we will compare their performances on never seen data in order to understand if they behave on test data the same way they did during this validation phase. We will also test other models to have a complete analysis.

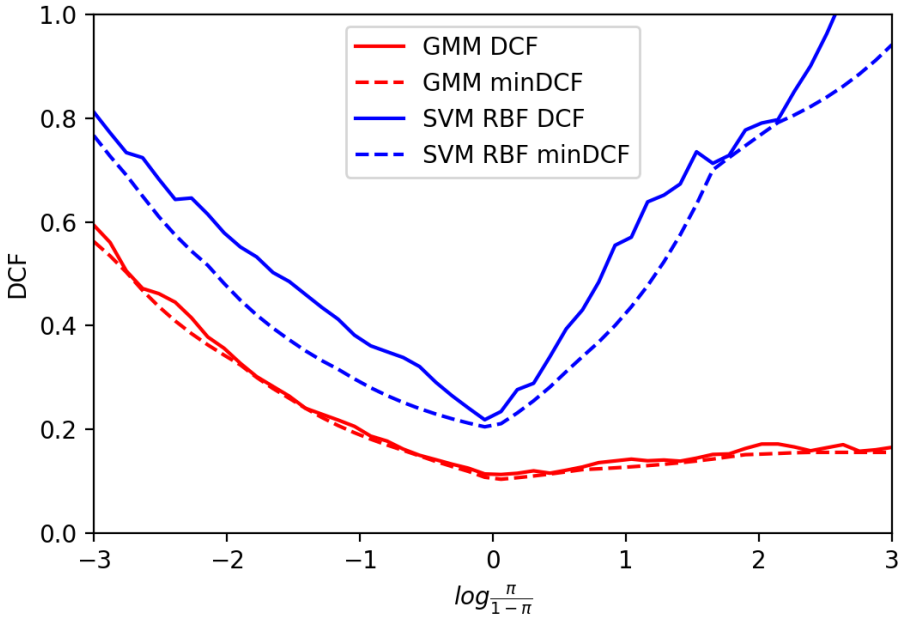
4 Evaluation phase

We now analyze performance on the evaluation set. In this phase models are trained on training data with a single pass (no k-fold) and tested on evaluation data.

We start from the previous phase selected models, reporting minimum and actual C_{prim} costs, without showing the results for each target working point:

Model	Validation set		Evaluation set	
	min C_{prim}	act C_{prim}	min C_{prim}	act C_{prim}
GMM 1-F 64-T, Z-Norm	0.185	0.204	0.237	0.250
RBF SVM $\gamma = 0.1$, $C = 10$, $K = 0.01$, PCA 5	0.099	0.118	0.365	0.417

also Bayes error plot is reported:



Bayes plot evaluation set

GMM results are similar to the ones computed during validation phase, on contrary for SVM RBF results show that performances are characterized by miscalibration. Latter model performed really worse on evaluation set than on validation set, this we think could be an example of overfitting because the model is able to perform well on training data but not on evaluation data, which is composed by totally new samples.

We now analyze our previous choices (models and configurations) to assess whether we obtained optimal or close to optimal results. Since linear classifiers are not suited to the task, we do not consider these models anymore. The comparisons will be performed exploiting minimum C_{prim} as metric.

4.1 Gaussian Classifiers

Gaussian models results on evaluation set are reported now, specifically the following table contains the best result for each version of the model:

Model	PCA	min C_{prim}
Full	5	0.325
Naive Bayes	5	0.326
Tied	5	0.794

So, also on evaluation set, results show that this family of models is not the best option for our kind of data, but we didn't expect different scenario on test data.

4.2 Quadratic Logistic Regression

Quadratic Log-Reg (weighted) has been tested with different Prior π on evaluation set. For each tested prior the best computed result (with relative parameter λ) is reported in the table below for both validation and evaluation phase:

Prior π	Val λ	Val min C_{prim}	Eval λ	Eval min C_{prim}
Dataset	10	0.232	0.1	0.243
Fold	10	0.232	0.1	0.243
0.1	10	0.233	0.1	0.246
0.2	10	0.233	0.1	0.244
0.5	0.1	0.231	0.1	0.247

Quad Log-Reg best results on validation and evaluation set for each prior π without PCA

Fold Prior on evaluation set is equal to Dataset Prior since, as mentioned before, no k-fold is performed during model training in this phase.
We can extend the analysis also considering PCA for both the best performing configurations on the validation and evaluation set:

PCA	Best Validation Conf. Eval min C_{prim}	Best Evaluation Conf. Eval min C_{prim}
-	0.247	0.243
5	0.253	0.245
4	0.310	0.305
3	0.337	0.336

To conclude the analysis of Quadratic Log-Reg, we understood that the optimal configuration is Prior $\pi = Dataset$ and $\lambda = 0.1$ without PCA while our best configuration during validation phase was Prior $\pi = 0.5$ and $\lambda = 0.1$ without PCA. Results computed on evaluation set are comparable to the ones obtained with validation set, specifically they are worse than the one computed with our chosen GMM configuration but actually better than the one obtained with RBF SVM on evaluation set.

4.3 SVM

Linear SVM version has not been tested on evaluation set since, as declared before, linear models not being suitable for our data have not been considered in this phase. SVM Polynomial will also be skipped since it's similar to Quadratic Log-Reg.
SVM RBF, which was the best performing model in validation phase, will be analyzed now.
Results reported at the beginning of this chapter expose an evident difference in terms of performances between validation and evaluation phase. Usually this is a symptom of overfitting, for this reason we tested different combinations of hyperparameters and PCA on evaluation set. Best configurations and obtained results are reported below:

K	C	γ	PCA	min C_{prim}
0.01	C=0.1	0.01	5	0.204
0.1	C=10	0.001	5	0.201
0.01	C=0.1	0.01	5	0.207
0.01	C=10	0.001	5	0.207
0.01	C=10	0.001	-	0.210

The optimal configuration on evaluation set, reported in the second row of the previous table, during validation phase computed worse results than the configuration chosen from us, which was indeed the most performing with validation data.
We believe that overfitting happened in validation phase for our chosen configuration because of the γ value; despite parameter C being relatively high also in that case, probably γ was too large and this caused overfitting. The optimal configuration performs better probably because has a smaller value for parameter γ.

To conclude, reported results show that this model is able to achieve better performances in terms on minimum C_{prim} with configurations different from the one chosen from us in validation phase, for this reason our choice was sub-optimal.

4.4 GMM

Finally, let's analyze **GMM**. Considering that the chosen configuration in the validation phase was Full GMM for target class and Tied GMM for non-target class, different number of components have been tested and obtained results (that will not be reported to avoid redundancy) show that the chosen configuration of 1-F GMM and 64-T GMM with Z-Norm is the best possible in evaluation phase, exactly as it was in validation phase. We can extend also in this case the analysis applying PCA to this configuration and test it on evaluatiom data:

PCA	min C_{prim}
-	0.237
5	0.251
4	0.344
3	0.351

Consequently also the choice performed in validation phase regarding PCA is optimal in this case.
Moreover, different combinations of GMM versions have been tested and for each combination the best result on evaluation set is reported in the table below:

GMM	Z-Norm	PCA	min C_{prim}
1-F 64-T	Yes	-	0.237
1-F 64-D	No	5	0.241
1-D 64-T	Yes	-	0.237
1-D 32-F	Yes	-	0.244
1-T 64-D	No	5	0.241
1-F 32-F	Yes	-	0.244

Equivalent configurations are not reported to avoid redundancy, for example the 1-T 64-T is exactly the same as 1-F 64-T. Table results confirm that the configuration chosen in validation phase is an optimal choice.

5 Conclusions

In the following table GMM and SVM RBF models, which are the ones selected in validation phase from us, are compared. Specifically, for each model minimum and actual C_{prim} values computed with chosen and optimal configuration on evaluation set are reported:

Model	Chosen conf.		Optimal conf.	
	min C_{prim}	act C_{prim}	min C_{prim}	act C_{prim}
RBF SVM	0.365	0.417	0.201	0.231
GMM	0.237	0.250	0.237	0.250

Summarizing, results show that our choice was sub-optimal in the SVM case because chosen configuration doesn't perform well on evaluation data, while the optimal configuration performs better than any other. In the GMM case the chosen configuration is the optimal one also on evaluation set as analysis performed in previous chapter shown, but its results are not as good as the ones computed with the optimal SVM configuration.

We can conclude that our choices led to optimal results with GMM model, and sub-optimal results in the SVM case which was the model selected as "final" from us at the end of validation phase.