Gender identification

Masoud Karimi - 300283

Polytechnic of Turin

01URTOV : Machine Learning and Pattern Recognition

1 Introduction

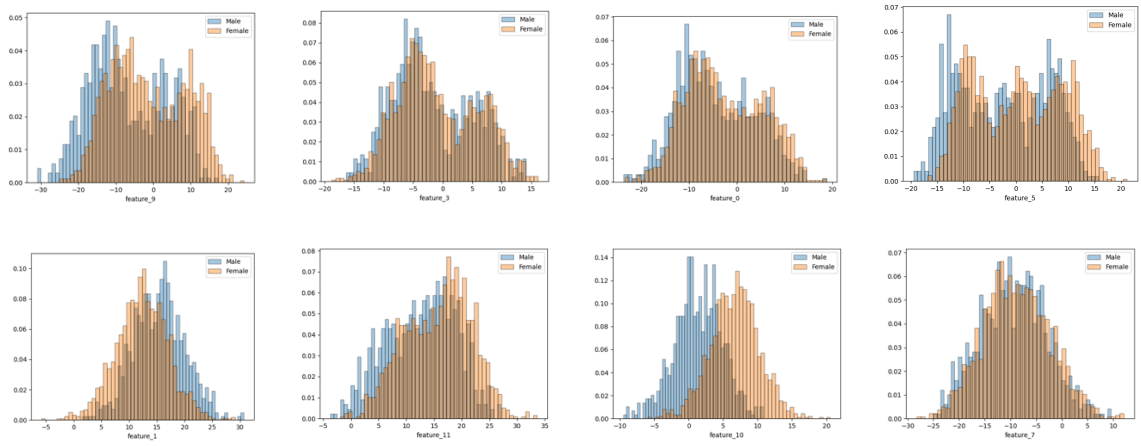
The project aims to develop a classifier that can determine the gender of individuals using high-level features extracted from facial images. Such tools are useful, for instance, as a preprocessing step for gender-specific face recognition models. The dataset consists of image embeddings, i.e., low-dimensional representations of images obtained by mapping face images to a common, low-dimensional manifold (typically few hundred dimensions), for example by means of suitable neural networks. To keep the model tractable, the dataset consists of synthetic data, and embeddings have significantly lower dimension than in real use-cases.

The embeddings are 12-dimensional continuous-valued vectors, categorized as either male or female. It is important to note that the individual components of the embeddings do not hold any direct physical interpretation. In addition, the training dataset comprises 1680 female samples and 720 male samples. On the other hand, the evaluation dataset consists of 1800 female samples and 4200 male samples.

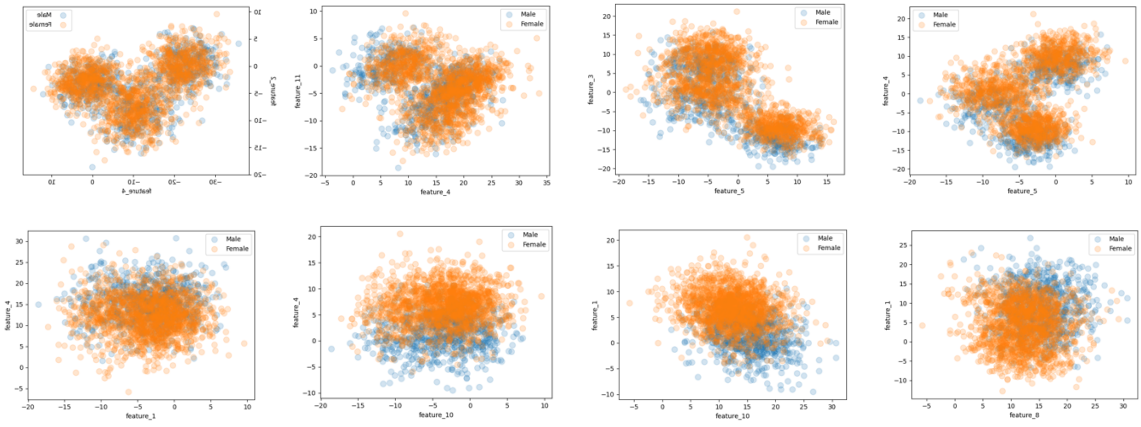
Our main application will be a uniform prior one and uniform cost for different misclassifications, with a working point defined by the triplet (π_T=0.5, Cfn=1, Cfp=1). In addition, we will also consider unbalanced applications where the prior is biased towards one of the two classes, (π_T=0.9, Cfn=1, Cfp=1) and (π_T=0.1, Cfn=1, Cfp=1).

2 Features

After conducting an initial examination of the training data, specifically through the analysis of histogram plots, it appears that the raw features demonstrate a distribution that might resemble a Gaussian distribution. However, it is important to note that certain histograms exhibit a bimodal shape, indicating the potential presence of underlying modes within the data. This observation challenges the assumption of a purely Gaussian distribution, as the data might be better characterized by a mixture of Gaussian distributions or other non-Gaussian models.

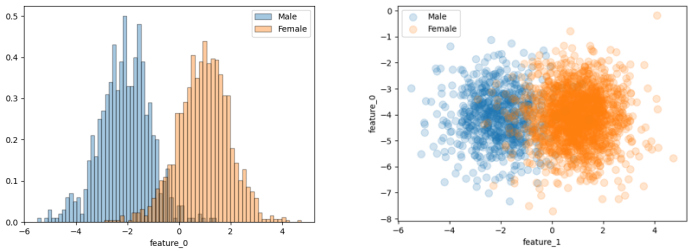


Additionally, the analysis of scatter plots provides good evidence for the presence of distinct patterns within the classes. The observed distinct patterns suggest that the underlying data within each class likely deviates from a purely Gaussian distribution and/or may exhibit more complex structures. It is important to emphasize that employing marginals in plotting does not ensure that the point density of all dimensions follows a Gaussian distribution.

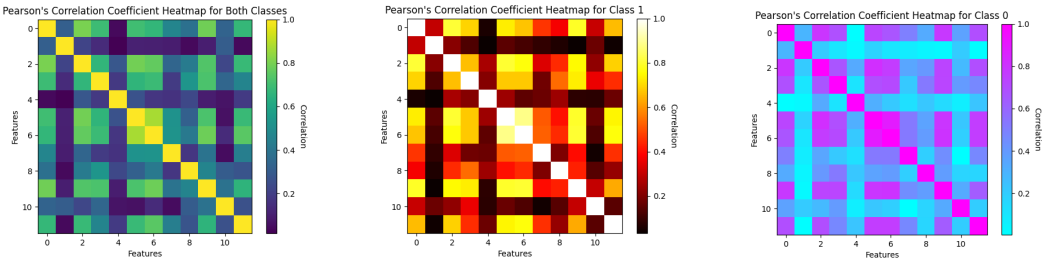


Based on the observed distribution of features in the scatter plots, it is anticipated that non-Gaussian and/or non-linear models would be considerably more effective for the given task.

By analyzing the histogram of the most discriminant direction obtained from LDA, we can gain valuable insights into the potential effectiveness of linear classifiers. The histogram analysis suggests that a linear classifier has the capability to reasonably separate the classes. However, it is important to emphasize that there is still a possibility of making some mistakes during the classification process due to existing overlap.



showing the absolute value of the Pearson correlation coefficient for both class(left image), for samples belong to female class (middle image) and samples belong to male class(right image)



Through the analysis of the correlation matrix, it is evident that features 6 and 5 exhibit a strong correlation. This indicates that employing PCA to transform the data into 11 uncorrelated directions might be beneficial. However, it is important to note that when the number of dimensions is limited, PCA may inadvertently remove valuable information. Moreover, PCA is a linear transformation it might not work well, this will assess if applying PCA is useful or not. Furthermore, the presence of correlations between features for each class, as indicated by the heatmap, suggests that Naive Bayes assumption of feature independence may result in inaccurate predictions. Maybe applying PCA be helpful in this case

Moreover, upon analyzing the covariance matrix of each class, it becomes apparent that they share a notable degree of similarity. This finding suggests that employing a linear classifier like tied Gaussian classifier has the potential to create a strong separation rule, aligning with our earlier discussion on the analysis of the leading direction of LDA.

3 Train Protocol

The performance of each model is measured in terms of minimum costs, specifically using minDCF (minimum Detection Cost Function). To determine the most promising model and evaluate the impact of using PCA, a K-fold cross-validation approach with K=3 is adopted. This choice helps mitigate the time consumption typically associated with the leave-one-out approach. By utilizing K-fold cross-validation, the performance of different models can be assessed more efficiently, providing valuable insights into their effectiveness. This approach allows for an efficient evaluation of the performance at a specific working point with different hyperparameters.

4 Gaussian Classifiers

model	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Full Covariance		Raw Feature	
	0.113	0.306	0.352
		PCA 11	
	0.119	0.301	0.364
		PCA 10	
	0.167	0.426	0.484
Tied Covariance		Raw Feature	
	0.111	0.304	0.337
		PCA 11	
	0.116	0.291	0.352
		PCA 10	
	0.161	0.401	0.463
Diag. Covariance		Raw Feature	
	0.466	0.782	0.785
		PCA 11	
	0.128	0.327	0.363
		PCA 10	
	0.166	0.448	0.467

PCA appears to be ineffective overall. As predicted by the analysis of the heatmap plot, the Naive Bayes model with raw features performs worse compared to other models with

raw features. However, performing PCA shows promise for this specific model as it maps the data to a space where the directions are uncorrelated.

Furthermore, based on the analysis of covariance matrices of classes and leading directions of LDA, it is evident that the tied model achieves the best performance in the K-fold cross-validation protocol, both with and without PCA, when compared to other models. When comparing the full covariance model with the tied model, it is evident that the linear decision surface performs better compared to the quadratic one.

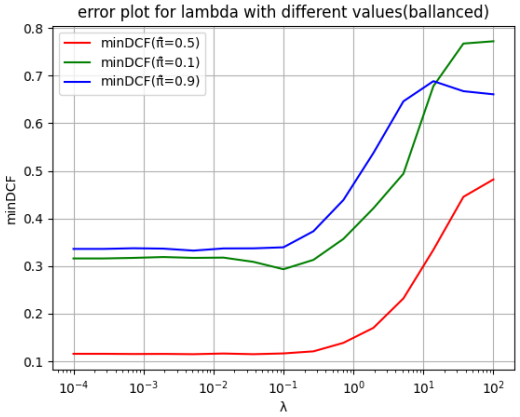
It is also noticeable that unbalanced applications yield worse results in comparison to balanced one.

5 Logistic regression

We analyze regularized (linear) Logistic Regression and take into account the imbalanced distribution of classes. To address this issue, we rebalance the costs of the different classes by minimizing an objective function that incorporates various empirical priors, π_T, through a prior weighted version of the model.

To initiate our analysis of the primary application, we assign the value of π_T=0.5. Consequently, we can evaluate and compare the models by employing various values of λ.

We initiate the analysis of the linear classifier without utilizing PCA. However, it appears that regularization is not particularly effective.



In addition, we have the option to explore training using a different prior, π_T, and observe its impact on other applications. For our analysis, we focus on models that incorporate minimal regularization based on previous plot of different values of lambda.

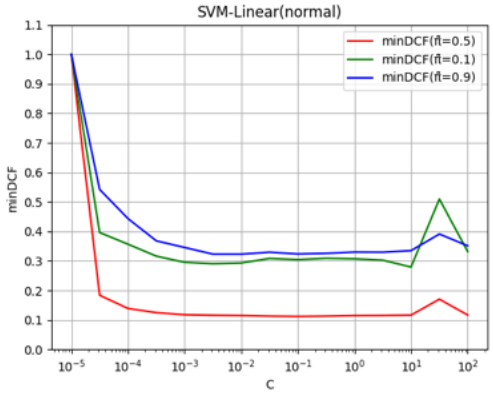
model	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Linear Logistic Regression(π_T =0.5, λ= 1e-4)	Raw Feature		
	0.113	0.308	0.336
	PCA 11		
	0.118	0.292	0.353
	PCA 10		
	0.167	0.402	0.468
Linear Logistic Regression(π_T =0.1, λ= 1e-4)	Raw Feature		
	0.119	0.302	0.396
	PCA 11		
	0.123	0.302	0.371
	PCA 10		
	0.164	0.400	0.504
Linear Logistic Regression(π_T =0.9, λ= 1e-4)	Raw Feature		
	0.115	0.316	0.336
	PCA 11		
	0.117	0.325	0.350
	PCA 10		
	0.158	0.423	0.461

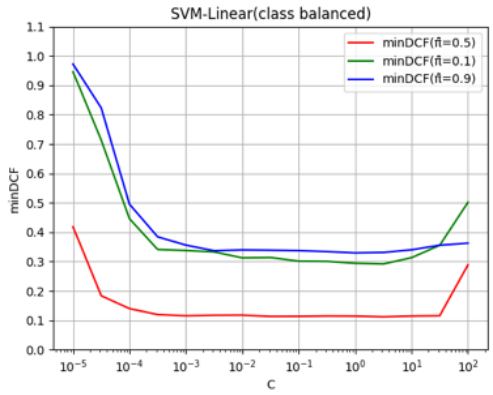
Overall, the Multivariate Gaussian (MVG) model with tied covariances demonstrates superior performance. Logistic regression, being a linear classifier, provides relatively similar results to the tied covariances model. Experimenting with different values of πT does not yield improvements in the Logistic Regression models for the other two applications.

Based on our findings from generative models and the results obtained from linear logistic regression, it is evident that applying PCA does not yield significant benefits. Therefore, for the remaining analysis, we will solely consider the entire set of features without utilizing PCA.

6 Support Vector Machines

The significance of the choice of C for linear SVM (without class balancing) can be observed from the left plot and the right plot showcasing the class balancing. When considering various values of C, it becomes evident that the selection of C does not appear to be crucial. Hence, C = 0.1 is selected.

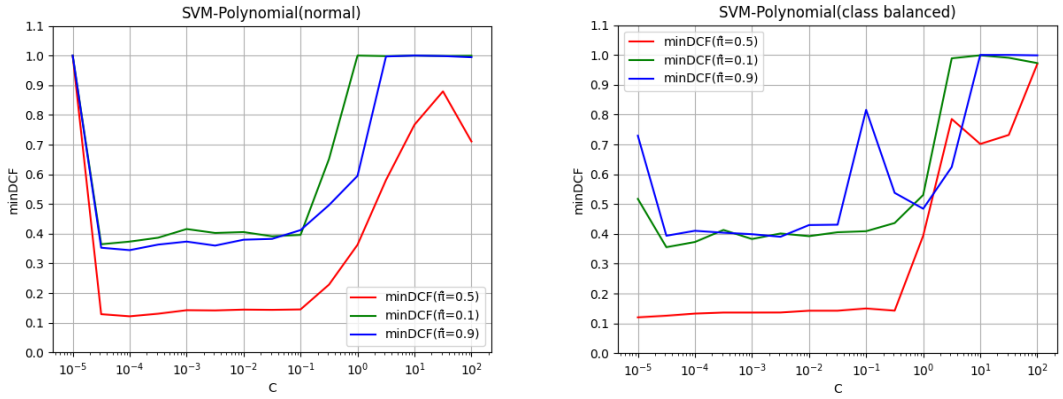




Model	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
		Raw Feature	
MVG(Tied Full-Cov)	0.111	0.304	0.337
Log Reg(π_T =0.5, λ= 1e-4)	0.113	0.308	0.336
Linear SVM(C=0.1)	0.112	0.304	0.323
Linear SVM(C=0.1, π_T =0.5)	0.113	0.293	0.344
Linear SVM(C=0.1, πT =0.1)	0.122	0.301	0.382
Linear SVM(C=0.1, πT =0.9)	0.116	0.326	0.337

By comparing linear models based on the minDCF, we observe that the performance of Linear SVM is approximately similar to other linear approaches. It is evident that class balancing is ineffective for linear SVM. However, when using the kernel version of SVM, it is essential to examine whether class balancing provides any advantages.

Now, let's shift our focus to the quadratic kernel SVM. Once more, it is clear that the selection of C is not crucial within the range of [1e-4, 1e-1]. To be cautious and minimize risks, other intervals are included, although they do not show promising results, as depicted in the plot. In this scenario, C=1e-5 is chosen as it seems to be a suitable option for class balanced and C=1e-4 for normal objective function . Furthermore, it is apparent that class balancing marginally enhances the outcomes.

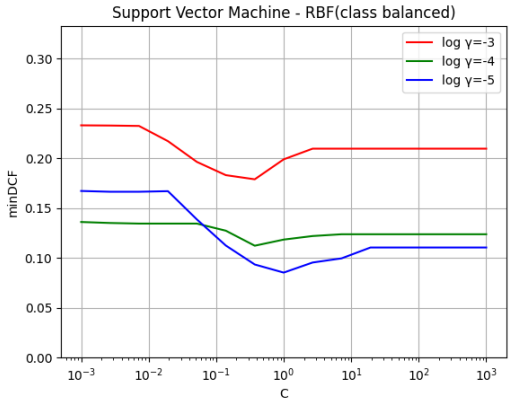


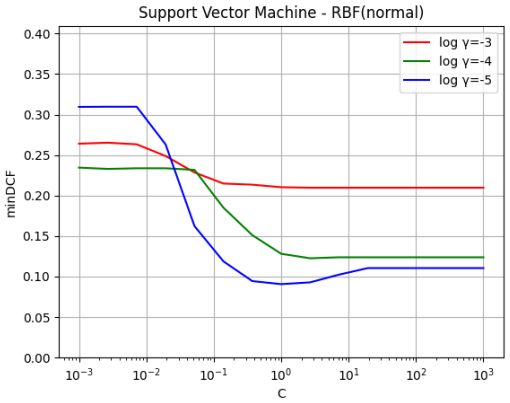
By evaluating the min-DCF, we can compare quadratic models. It is apparent that the Quadratic kernel SVM yields slightly inferior results compared to the MVG Full-Cov model. The diminished performance can be attributed to class imbalance. However, when retraining the Quadratic kernel SVM with the estimated C value while incorporating class balancing, we obtain slightly improved outcomes. Nevertheless, the MVG Full-Cov model obtains better results yet.

Model	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
		Raw Feature	
MVG(Tied Full-Cov)	0.113	0.306	0.352
Poly(C=1e-4)	0.121	0.373	0.344
Poly(C=1e-4, π_T =0.5)	0.132	0.367	0.381
Poly(C=1e-5, π_T =0.5)	0.120	0.345	0.353

The grid search plot for RBF kernel SVM illustrates the minimum DCF (π̃=0.5) achieved with different combinations of C and γ. The left plot represents the results with class balancing, while the right plot shows the results without class balancing. The plot clearly demonstrates that both γ and C have an impact on the outcomes. Moreover, optimizing both parameters together is crucial, as the optimal C value depends on the chosen γ.

The grid search reveals that the best results, in both class balanced and normal scenarios, are obtained with C=1.0 and log(γ)=-5. Although other values of γ were evaluated, they produced inferior results and are therefore not reported. The best results achieved on the validation set were obtained using a class-balanced SVM classifier with a RBF kernel and the raw features. It is worth noting that class balancing proves to be useful in this case.





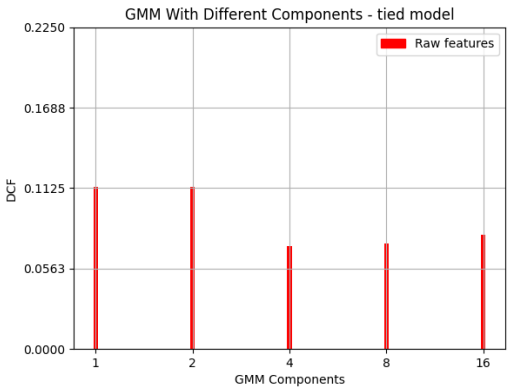
Model	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
		Raw Feature	
RBF(C=1.0, log(γ)=-5)	0.091	0.289	0.264
RBF (C=1.0, log(γ)=-5, π_T =0.5)	0.086	0.265	0.266

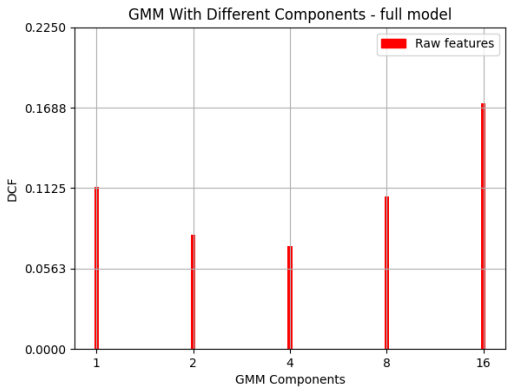
7 Gaussian Mixture Model

Due to the capability of Gaussian Mixture Models to approximate various types of distributions, and considering our previous analysis of the data distribution, we anticipate obtaining improved results compared to the Gaussian model. Both full covariance and tied models are considered in analysis.

The following bar plots depict the min-DCF($\tilde{\pi} = 0.5$) for both full and tied covariance models, respectively right and left plots. Based on the analysis of these bar plots, it is evident that the optimal results were achieved for both tied and full models when utilizing four components for each class. Additionally, both models exhibited remarkably similar performance when employing four components. Furthermore, it should be noted that an increase in the number of components leads to a higher DCF due to overfitting.

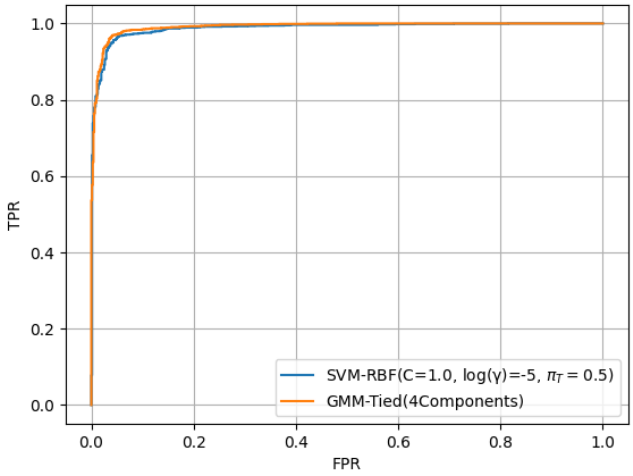
It is worth mentioning that to prevent the covariance matrices from shrinking towards zero, a constraint is applied to enforce minimum values for their eigenvalues in this project 0.01. This constraint ensures that the covariance matrices retain a certain level of variation and do not become overly small. By imposing this constraint, the model avoids degeneracy issues and maintains the necessary diversity in the covariance matrices.





Model	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
		Raw Feature	
RBF (C=1.0, log(γ)=-5, π_T =0.5)	0.0862	0.2653	0.2661
GMM- Full-Cov(4Componets)	0.0722	0.2190	0.1920
GMM- Tied-Cov(4Componets)	0.0720	0.2494	0.2291

The ROC plot provides evidence that the GMM model outperforms or performs equally to the SVM model at all operating points. The GMM Tied-Cov model, utilizing four components, is chosen as the candidate model. As a secondary system, the SVM model with an RBF kernel, trained with balanced classes (π_T=0.5), C=1.0, and log(γ)=-5, is selected.

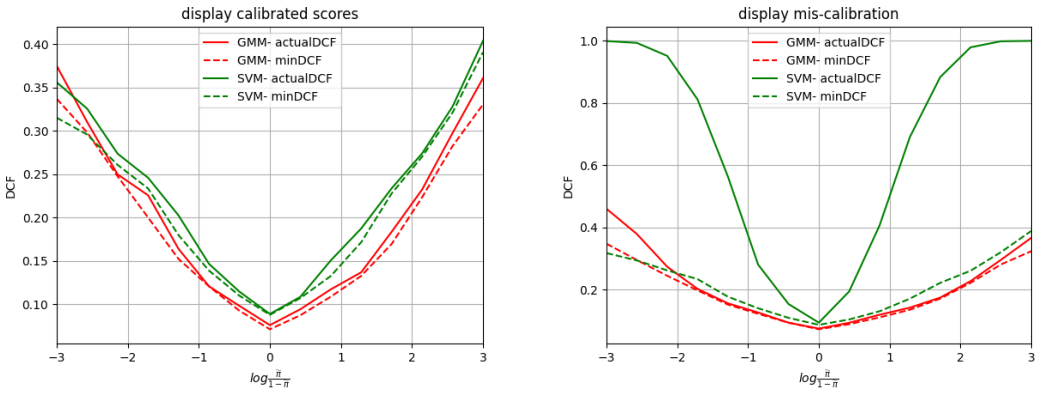


The min-DCF evaluates the cost associated with making optimal decisions for the evaluation set based on the scores provided by the recognizer. However, the actual cost depends on the quality of the decisions made using those scores, specifically on the effectiveness of the threshold used in practice to assign classes in the binary case. Hence, we shift our focus to the actual DCFs to assess the real costs involved.

	$\tilde{\pi} = 0.5$		$\tilde{\pi} = 0.1$		$\tilde{\pi} = 0.9$	
	min-DCF	act-DCF	min-DCF	act-DCF	min-DCF	act-DCF
GMM	0.0720	0.0742	0.2494	0.2875	0.2291	0.2375
SVM	0.0862	0.0932	0.2653	0.9625	0.2661	0.9819

It is apparent that the GMM yields scores that are mostly well-calibrated across our three applications. However, provided scores by the SVM model, despite having a probabilistic interpretation, for primary application, are almost calibrated, but for other applications scores are mis-calibrated. This is confirmed by a Bayes error plot, which shows the DCFs for different applications(right hand-side plot). It is apparent that the SVM yields well-calibrated scores only for one specific application where the log-odds of the effective prior is equal to 0, such as when the effective prior is 0.5. However, for the remaining applications, the scores provided by the SVM are significantly mis-calibrated.

To recalibrate both models, a prior-weighted logistic regression approach is utilized. After the calibration process(left hand-side plot), it becomes evident from the plot that the scores are properly calibrated for almost all applications for both models.



The table presents both the minimum and actual costs associated with the calibrated scores.

	$\tilde{\pi} = 0.5$		$\tilde{\pi} = 0.1$		$\tilde{\pi} = 0.9$	
	min-DCF	act-DCF	min-DCF	act-DCF	min-DCF	act-DCF
GMM	0.0712	0.0759	0.2529	0.2589	0.2317	0.2444
SVM	0.0888	0.0878	0.2648	0.2833	0.2765	0.2831

8 Evaluation

Once again, we begin with the Gaussian classifiers and assess the systems based on their minimum DCFs. This approach ensures that we can confirm whether the proposed solution is truly capable of achieving the highest level of accuracy.

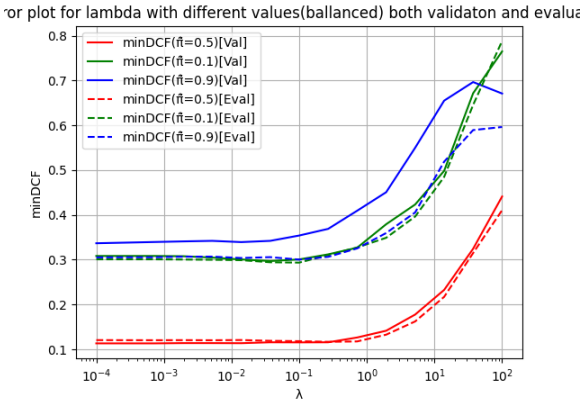
The results obtained on the validation set align with the previous findings. It is evident that the MVG model with tied covariance over raw features is the most suitable choice for our primary application. The similarity in results suggests that the training and evaluation populations in this specific use case share similar characteristics. Furthermore, it is evident that, as observed in the validation set, the application of PCA is ineffective.

model	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Full Covariance		Raw Feature	
	0.119	0.312	0.312
		PCA 11	
	0.123	0.321	0.332
		PCA 10	
	0.154	0.409	0.428
Tied Covariance		Raw Feature	
	0.116	0.301	0.308
		PCA 11	
	0.121	0.314	0.322
		PCA 10	
	0.150	0.402	0.419
Diag. Covariance		Raw Feature	
	0.434	0.817	0.705
		PCA 11	
	0.130	0.356	0.323
		PCA 10	
	0.160	0.433	0.422

Next, we examine linear logistic regression models with the estimated value of λ = 10−4. Once again, the results align with our expectations. The linear models demonstrate comparable performance to the MVG linear classifier. Again, it is clear that applying PCA is ineffective, and the best results are achieved when using raw features. As anticipated, the π_T =0.5 yields the optimal outcome in terms of minimum DCF.

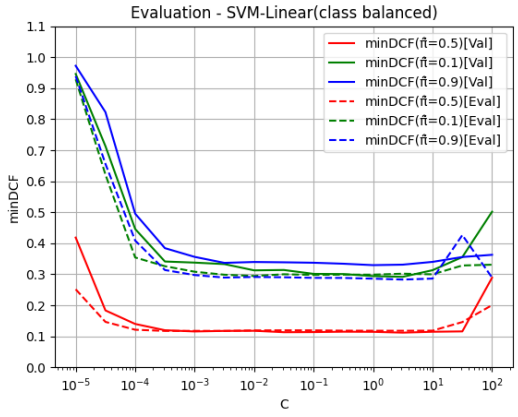
model	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Linear Logistic Regression(π_T =0.5, λ= 1e-4)	Raw Feature		
	0.120	0.300	0.305
	PCA 11		
	0.124	0.314	0.318
	PCA 10		
	0.151	0.404	0.417
Linear Logistic Regression(π_T =0.1, λ= 1e-4)	Raw Feature		
	0.122	0.296	0.339
	PCA 11		
	0.123	0.307	0.338
	PCA 10		
	0.159	0.400	0.445
Linear Logistic Regression(π_T =0.9, λ= 1e-4)	Raw Feature		
	0.123	0.340	0.279
	PCA 11		
	0.128	0.347	0.307
	PCA 10		
	0.156	0.422	0.411

While we do not anticipate significant deviations, it is worth verifying whether the selected λ value produces results close to optimal. The curves for both the validation and evaluation sets exhibit a similar trend, indicating the effectiveness of our choice, despite the overall modest performance of the linear model.



We start with the linear SVM approach. Similarly, we evaluate models trained using the estimated value of C obtained from the K-fold validation set. In this case we can see that in contrary with the validation phase, class balanced model obtains better result. We can see that selected C value is good enough based of following plot.

Model	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
		Raw Feature	
Linear SVM(C=0.1)	0.122	0.321	0.287
Linear SVM(C=0.1, π_T =0.5)	0.119	0.307	0.294



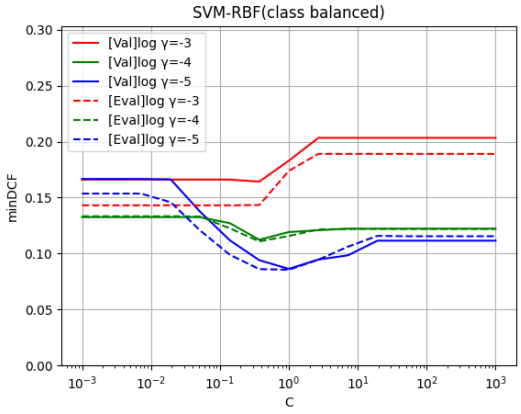
Now we consider the polynomial SVM approach. Similarly, we evaluate models trained using the estimated value of C obtained from the K-fold validation set. Contrary to our findings in the validation phase, the polynomial SVM demonstrates superior performance compared to linear models. The class-balanced model utilizing raw features achieves the best results on the evaluation set.

Model	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
		Raw Feature	
Poly(C=1e-4)	0.1114	0.3111	0.2671
Poly(C=1e-4, π_T =0.5)	0.1055	0.3135	0.2890
Poly(C=1e-5, π_T =0.5)	0.1130	0.3265	0.3021

After considering the models trained with the values of C and γ estimated on the K-fold validation set for SVM with RBF kernel, we observe that our results align with those from the validation phase. Once again, the class-balanced model utilizing raw features, where π_T =0.5, achieves the best results on the evaluation set.

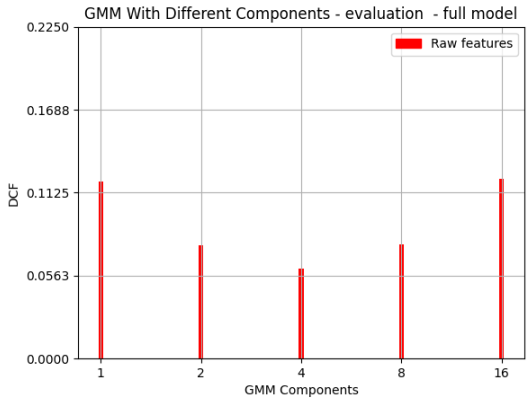
Model	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
		Raw Feature	
RBF(C=1.0, log(γ)=-5)	0.092	0.284	0.195
RBF (C=1.0, log(γ)=-5, π_T =0.5)	0.085	0.246	0.222
RBF (C=1.0, log(γ)=-5, π_T =0.1)	0.992	0.235	0.273
RBF (C=1.0, log(γ)=-5, π_T =0.9)	0.108	0.377	0.203

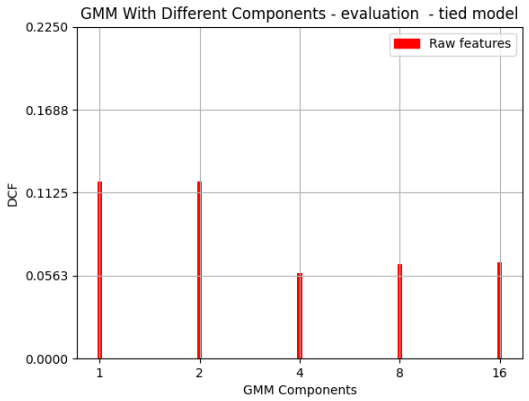
The impacts of utilizing different values of C and γ remain consistent with our observations on the validation set. Hence, our selection of γ and C has proven to be effective.



Let's discuss the GMM classifiers now. Once again, we focus on minimizing the DCF on the evaluation set, which consists of models trained with varying numbers of Gaussians. Specifically, we compare the performance of a model with full covariance against one with tied covariance. The evaluation results align consistently with the validation results.

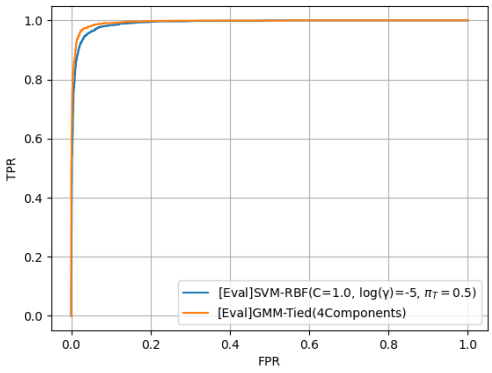
In terms of the optimal number of components, both the evaluation and validation sets show similar findings for both models. Notably, when we utilize the entire dataset, we observe a significant improvement in accuracy.



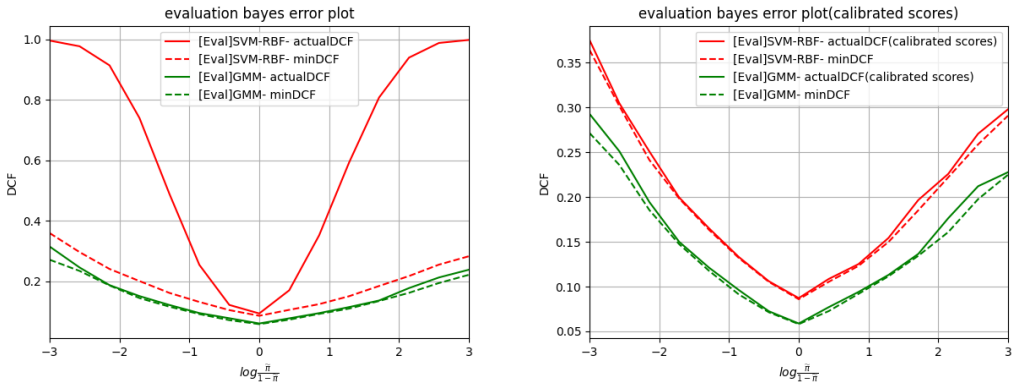


Model	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
		Raw Feature	
GMM- Full-Cov(4Componets)	0.061	0.155	0.155
GMM- Tied-Cov(4Componets)	0.057	0.191	0.165

As mentioned during the validation phase, the ROC plot provides compelling evidence that the GMM model performs better or at least equal with the SVM model across all operating points.



It can be seen that the GMM scores are well-calibrated



In conclusion, the GMM approach with Tied-Covariance(four components) proves to be effective and generates well-calibrated scores that can be applied to various scenarios. We have achieved a DCF cost of approximately 0.06 for our designated working point with a prior proportion of 0.5.

However, it is worth noting that these models may not perform as effectively when faced with applications that involve imbalanced costs and prior proportions.

The overall similarity observed between the validation and evaluation results suggests that the evaluation population is adequately representative of the training population. Thus, the choices we made regarding the training and validation sets have proven to be effective for evaluating the performance of the models on the evaluation data.