Wine quality project

Luca Stradiotti 283340, Elisa Tedde 286282

1. Introduction

The project analyzes different classification techniques (MVG, LR, SVM and GMM) in a binary task. The problem is divided into different stages. Firstly, different models have been analyzed in order to describe relationships between features and labels of the training set. In this step model parameters and hyperparameters have been computed. The final step consists in comparing the results and the predictions on the validation set with the performances obtained on the evaluation set in order to assess the goodness of taken decisions.

2. Dataset

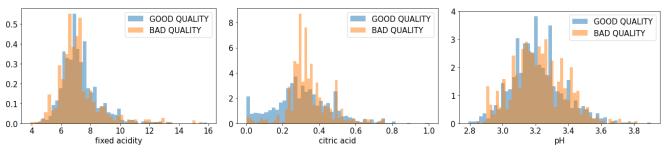
The project is aimed at discriminating between good and bad quality wines. The two datasets are related to red and white variants of the Portuguese "Vinho Verde" wine. The original dataset consists of 10 classes, rating quality from 1 to 10, but it has been binarized, grouping all wines with low quality (lower than 6) into class 0, and all those with good quality (greater than 6) into class 1; wines with quality grade 6 have been discarded to simplify the task. The dataset has already been split into training and test set. The training set is composed of 1839 instances, including 613 good and 1226 bad wines; the test set contains 1822 samples instead, of which 664 represents good wines, whilst the others bad ones. The output is based on wine quality.

2.1. Features analysis

There are 11 features, that represent physical properties of the wine. Input variables are all continuous:

1. fixed acidity.
2. volatile acidity
3. citric acid
4. residual sugar
5. chlorides
6. free sulfur dioxide
7. total sulfur dioxide
8. density
9. pH
10. sulphates
11. alcohol

The last column represents the label.



(a) (b) (c)

Figure 1: examples of distributions: (a) fixed acidity (b) citric acid (c) pH

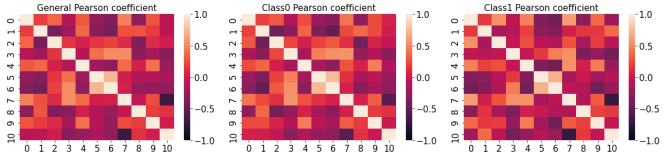
A preliminary analysis of the training set using histograms to visualise the distribution of all the attributes for two classes demonstrates that raw features largely overlap, as the dataset is composed of a higher number of medium quality wines than excellent or poor ones. Moreover, the raw features show irregular distributions with large outliers. Likely leading classification approaches to yield sub-optimal results.

2.2. Heat Map

The heat map has been used in order to analyze the correlation between the features. The Pearson correlation coefficient was calculated with the formula

$$\frac{Cov\,(X, Y,)}{\sqrt{Var\,(X)}\sqrt{Var\,(Y)}}$$

and it is visualized through the correlation matrices.



(a) (b) (c)

Figure 2: (a) whole dataset, (b) bad wines, (c) good wines

Graphs shows that features 5 and 6 are strongly correlated, as well as features 7 and 10. Even though the input space is composed of 11 features only, PCA may be useful to improve classification.

3. Model

3.1. K-Fold cross-validation

K-Fold cross-validation has been used to understand which model is most promising. In this case the approach involves randomly dividing the dataset into 5 folds, of approximately equal size. One iteration is made for each fold: selected fold is treated as validation set while the remaining $k-1$ folds as training set. The final classifier will be obtained by re-training over the whole set, so it will leverage additional data. Decisions are made over the validation set.

3.2. Gaussianization

Gaussianization is a procedure that allows to map a set of features to values whose empirical cumulative distribution function is well approximated by a Gaussian distribution. It is applied both on the training and evaluation samples. The first step consists in mapping the features into a uniform distribution and subsequently

transforming with through the inverse of the cumulative distribution function.

$$r(x) = \frac{\sum_{i=1}^{N} \mathbb{I}\left[x < x_i\right] + 1}{N + 2}$$

Rank is computed considering that x is the feature to be transformed while $\mathbb{I}$ is the indicator function. To avoid numerical issues, it is convenient to add 1 to the numerator and 2 to the denominator. Afterwards ppf (percent point function) is applied to the computed rank to obtain a standard normal distribution.

3.3. Working points

The triplet $(\tilde{\pi}, C_{fp}, C_{fn})$ represents the working point of an application for a binary classification task. The main application that will be considered has a uniform prior.

$$(\tilde{\pi}, C_{fp}, C_{fn}) = (0.5, 1, 1)$$

Also unbalanced applications will be considered, where the prior is biased towards one of the classes.

$$(\tilde{\pi}, C_{fp}, C_{fn}) = (0.7, 1, 1) \quad (\tilde{\pi}, C_{fp}, C_{fn}) = (0.1, 1, 1)$$

In order to choose the most promising approach, the performances will be measured in tearms of normalized minimum detection costs, which represents the cost that is paid knowing before-hand the optimal threshold for the evaluation and it is used to measure the perfomance of the application.

3.4. Gaussian classifier

The first classification method is the Gaussian classifier. Both raw and gaussianized features are considered. Four different Gaussian classifiers are analyzed: full covariance, diagonal covariance, tied full coviarance and tied diagonal covariance.

	Raw			**Gaussianization**		
	$\widetilde{\pi_1}$	$\widetilde{\pi_2}$	$\widetilde{\pi_3}$	$\widetilde{\pi_1}$	$\widetilde{\pi_2}$	$\widetilde{\pi_3}$
no PCA						
Full cov	0.412	0.644	0.831	0.404	0.611	0.847
Diag cov	0.454	0.790	0.852	0.489	0.829	0.866
Tied full cov	**0.349**	0.543	0.867	0.390	0.526	0.852
Tied diag cov	0.420	0.673	0.885	0.470	0.756	0.864
PCA m= 10						
Full cov	0.368	0.574	0.837	0.397	0.617	0.848
Diag cov	0.462	0.750	0.902	0.474	0.749	0.896
Tied full cov	0.358	0.541	0.860	0.390	0.534	0.837
Tied diag cov	0.367	0.581	0.852	0.405	0.563	0.826
PCA m= 9						
Full cov	0.356	0.530	0.847	0.396	0.632	0.852
Diag cov	0.458	0.703	0.883	0.477	0.718	0.893
Tied full cov	0.351	0.541	0.854	0.394	0.544	0.852
Tied diag cov	0.363	0.585	0.842	0.402	0.589	0.836

Table 1: minDCF for different Gaussian models: $\widetilde{\pi_1} = 0.5$, $\widetilde{\pi_2} = 0.7$, $\widetilde{\pi_3} = 0.1$

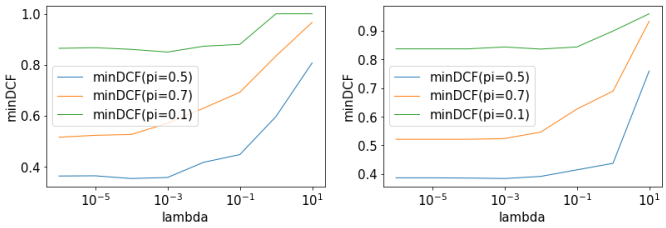
PCA is used to reduce the number of features, yet it does not improve the estimate since the number of features are limited. Gaussianization is not effective. The diagonal covariance model generally performs worse than full covariance models: Naive Bayes approach performs poorly because some features are correlated. Overall, the best candidate is tied full covariance that obtains the best performance as classes have similar distribution (covariance matrices are fairly similar).

3.5. Logistic regression

The second classification model analyzed is the logistic regression, a discriminative approach that does not require assumptions on the data distribution. Both raw and gaussianized features are considered. The hyper-parameter λ has to be selected using k-fold cross validation to optimize the performance of the classifier. PCA is not considered anymore, since it has limited effectiveness for generative models.

3.5.1 Linear logistic regression

The decision rules of linear logistic regression are linear surfaces orthogonal to w.



(a) (b)

Figure 3: minDCF for different values of λ: (a) raw features, (b) gaussianized features

Regarding the selection of λ, the best performance is obtained on raw features using $\lambda = 0.0001$ and on Gaussianized features with $\lambda = 0.001$.

Since classes are not balanced, different priors might be considered in order to re-balance the costs of the different class. The function that has to be minimized thus becomes:

$$J(\boldsymbol{w}, b) = \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{\pi_T}{n_T} \sum_{i=1|c_i=1}^{n} \log\left(1 + e^{-z_i\left(\boldsymbol{w}^T \boldsymbol{x}_i + b\right)}\right) + \\ + \frac{1 - \pi_T}{n_F} \sum_{i=1|c_i=0}^{n} \log\left(1 + e^{-z_i\left(\boldsymbol{w}^T \boldsymbol{x}_i + b\right)}\right)$$

	$\widetilde{\pi} = 0.5$	$\widetilde{\pi} = 0.7$	$\widetilde{\pi} = 0.1$
Raw features ($\lambda = 0.0001$)			
Linear LR ($\pi_T = 0.5$)	0.367	0.520	0.869
Linear LR ($\pi_T = 0.7$)	0.377	0.499	0.870
Linear LR ($\pi_T = 0.1$)	0.356	0.563	0.846
Linear LR	**0.354**	0.536	0.857
Gaussianized features ($\lambda = 0.001$)			
Linear LR ($\pi_T = 0.5$)	0.382	0.545	0.859
Linear LR ($\pi_T = 0.7$)	0.382	0.530	0.882
Linear LR ($\pi_T = 0.1$)	0.383	0.574	0.819
Linear LR	0.384	0.523	0.843

Table 2: minDCF for different class priors

Gaussianization does not provide improvements. For the primary application ($\widetilde{\pi} = 0.5$), the model with unbalanced classes

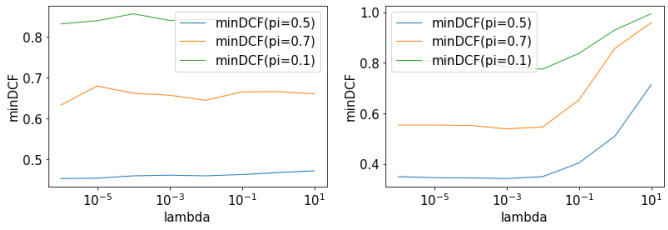
shows the best performance but it is not effective with respect to the tied full covariance one.

3.5.2 Quadratic logistic legression

This model finds quadratic separation rules since it uses as feature vectors

$$\phi(\boldsymbol{x}) = \begin{bmatrix} \text{vec}\left(\boldsymbol{x}\boldsymbol{x}^T\right) \\ \boldsymbol{x} \end{bmatrix}$$

rather than x: the decision rules are linear separation surfaces in the space defined by the mapping of ϕ.



(a) (b)

Figure 4: minDCF for different values of λ: (a) raw features, (b) gaussianized features

Regarding the selection of λ, the best performance is obtained on raw features using $\lambda = 0.000001$ and on Gaussianized features with $\lambda = 0.001$.

Again, class re-balancing might be considered.

	$\widetilde{\pi} = 0.5$	$\widetilde{\pi} = 0.7$	$\widetilde{\pi} = 0.1$
Raw features ($\lambda = 0.000001$)			
Quadratic LR ($\pi_T = 0.5$)	0.440	0.656	0.864
Quadratic LR ($\pi_T = 0.7$)	0.423	0.624	0.820
Quadratic LR ($\pi_T = 0.1$)	0.481	0.763	0.839
Quadratic LR	0.465	0.675	0.852
Gaussianized features ($\lambda = 0.001$)			
Quadratic LR ($\pi_T = 0.5$)	0.347	0.533	0.790
Quadratic LR ($\pi_T = 0.7$)	0.347	0.521	0.810
Quadratic LR ($\pi_T = 0.1$)	0.378	0.564	0.793
Quadratic LR	**0.342**	0.539	0.784

Table 3: minDCF for different class priors

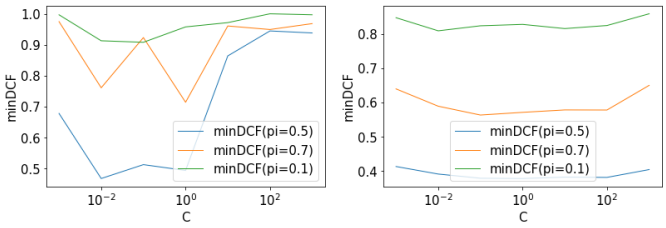
In this case, pre-processing seems to be more helpful than in the linear regression analysis. For the primary application ($\widetilde{\pi} = 0.5$), both models with un-balanced and balanced ($\pi_T = 0.5$) prior have a similar performance. Overall, quadratic logistic regression model trained with the empirical prior provides up to know the best performance.

3.6. SVM

The third classification approach analyzed is the SVM one. Both raw and gaussianized features are considered.

3.6.1 Linear SVM

The analysis starts considering linear SVM. For this model, the hyper-parameter C has to be tuned resorting to k-fold cross validation, this hyper-parameter allows selecting a trade-off between margin and errors on the training set. Initially, class rebalancing is not considered.



(a) (b)

Figure 5: minDCF for different values of C : (a) raw features, (b) gaussianized features

The choice of C looks critical for raw features, for which the best performance is achieved with C=0.01; conversely, as concerns the gaussianized ones, different values of C show quite similar performances. C=0.1 is selected.

Additionally, class rebalancing might be considered to improve performance. To rebalance the classes, different values of C must be used for each class.

$$C_T = C\frac{\pi_T}{\pi_T^{emp}} \qquad C_F = C\frac{\pi_F}{\pi_F^{emp}}$$

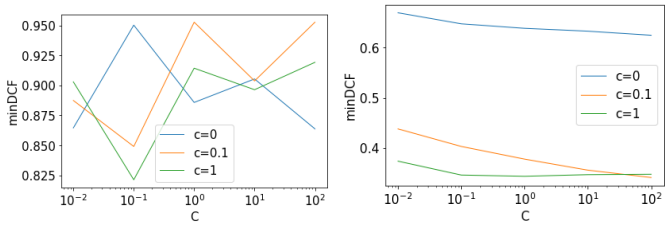
	$\widetilde{\pi} = 0.5$	$\widetilde{\pi} = 0.7$	$\widetilde{\pi} = 0.1$
Raw features ($C = 0.01$)			
Linear SVM ($\pi_T = 0.5$)	0.471	0.771	0.913
Linear SVM ($\pi_T = 0.7$)	0.502	0.780	0.986
Linear SVM ($\pi_T = 0.1$)	0.677	0.978	0.986
Linear SVM	0.467	0.761	0.913
Gaussianized features ($C = 0.1$)			
Linear SVM ($\pi_T = 0.5$)	**0.383**	0.578	0.839
Linear SVM ($\pi_T = 0.7$)	0.387	0.546	0.875
Linear SVM ($\pi_T = 0.1$)	0.887	0.999	1.000
Linear SVM	0.390	0.589	0.809

Table 4: minDCF for different class priors

The results show that in this case gaussianization improves the performance of the model and that class re-balancing slightly improves the performances. However, the results are quite poor with respect to quadratic logistic regression, probably because quadratic separation surfaces perform better for this dataset.

3.6.2 Kernel SVM

Additionally, two non-linear SVM formulations using kernel functions are examined. The first one uses a polynomial quadratic kernel, while the second employs a Radial Basis Function kernel. For the quadratic kernel, hyper-parameters C and c need to be estimated. A grid search to jointly optimize them on the primary application is used.



(a) (b)

Figure 6: minDCF ($\widetilde{\pi} = 0.5$) for different values of C and c: (a) raw features, (b) gaussianized features

The plot shows that both c and C influence the results: they should be optimized jointly. In this case, the model whit gaussianization performs significantly better than the one with raw features. Best results are obtained using $c = 0.1$ and $C = 100$.

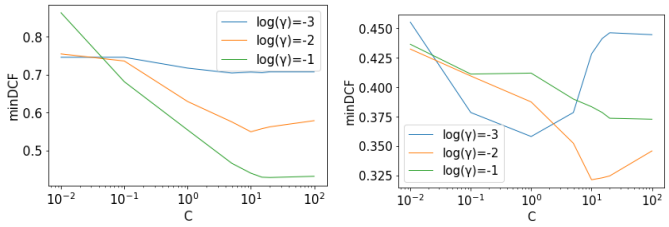
Gaussianized features ($C = 100, c = 0.1$)

	$\widetilde{\pi} = 0.5$	$\widetilde{\pi} = 0.7$	$\widetilde{\pi} = 0.1$
Quadratic SVM ($\pi_T = 0.5$)	0.333	0.507	0.781
Quadratic SVM ($\pi_T = 0.7$)	0.356	0.528	0.835
Quadratic SVM ($\pi_T = 0.1$)	0.412	0.620	0.800
Quadratic SVM	**0.329**	0.523	0.800

Table 5: minDCF for different class priors

For the primary application ($\widetilde{\pi} = 0.5$), the model with unbalanced classes performs slightly better than the one with balanced classes ($\pi_T = 0.5$), which, instead, gets better results for the other applications. Quadratic kernel SVM provides in general better results than quadratic logistic regression. So far, quadratic kernel SVM is the model that achieved the best performance.

Instead, γ and C should be jointly optimized for RBF kernel SVM.



(a) (b)

Figure 7: minDCF ($\widetilde{\pi} = 0.5$) for different values of C and γ: (a) raw features, (b) gaussianized features

Also in this case, gaussianized features performs significantly better than raw ones. The best model is obtained using $log(\gamma) = -2$ and $log(C) = 1$.

Gaussianized features ($C = 10, \gamma = 0.01$)

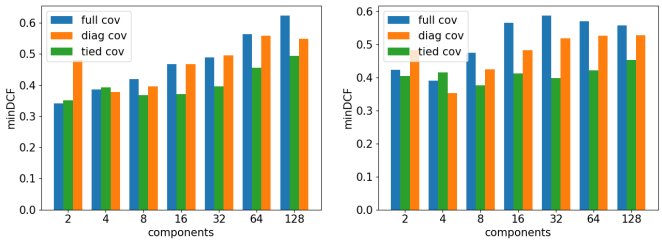
	$\widetilde{\pi} = 0.5$	$\widetilde{\pi} = 0.7$	$\widetilde{\pi} = 0.1$
RBF SVM ($\pi_T = 0.5$)	**0.316**	0.462	0.793
RBF SVM ($\pi_T = 0.7$)	0.319	0.456	0.760
RBF SVM ($\pi_T = 0.1$)	0.363	0.588	0.788
RBF SVM	0.321	0.470	0.783

Table 6: minDCF for different priors

Class re-balancing helps for different applications. So, it could be often useful training a task-specific classifier. Up to now, the RBF kernel SVM model trained with balanced classes and gaussianized features provides the best results on the validation set.

3.7. GMM

The last model to be examined is a generative approach based on training a GMM over the data of the two classes. GMMs can approximate generic distributions, hence they will probably provide better results than those obtained with the Gaussian model. Afterwards, full covariance, with and without covariance tying, and diagonal models are considered. For tied covariance model, tying is obtained at class level: different classes have different covariance matrices. Again, the selection of the most appropriate number of components is done through K-fold protocol.



(a) (b)

Figure 8: minDCF ($\widetilde{\pi} = 0.5$) for different values of C and γ: (a) raw features, (b) gaussianized features

The model that achieves the best performance is the full covariance one on raw features with two components for each class. However, the improvement compared to the gaussian classifiers is not very significant. All things considered, it does not improve the results obtained with the SVM with RBF kernel.

3.8. Score quality

Therefore, the candidate model are the SVM with RBF kernel trained with balanced classes ($\pi_T = 0.5$) on gaussianized features. Up to now, only minimum DCF metric has been considered, which measures the cost that would be paid if optimal decisions for the evaluation set were made using the recognizer scores. However, the cost depends on the goodness of the threshold used to perform class assignment. So, actual DCFs have to be considered to assess how the model performances would be using theoretical threshold for each application.

	$\widetilde{\pi}=0.5$		$\widetilde{\pi}=0.7$		$\widetilde{\pi}=0.1$	
DCF	min	act	min	act	min	act
RBF SVM	0.316	0.327	0.462	0.506	0.794	0.830

Table 7: minDCF and actDCF for different working points of the primary system

SVM with RBF kernel provides scores that are quite well calibrated for the primary application even though the model does not provide a probabilistic interpretation, while it may be beneficial to re-calibrate scores if the target application had un-balanced effective prior.

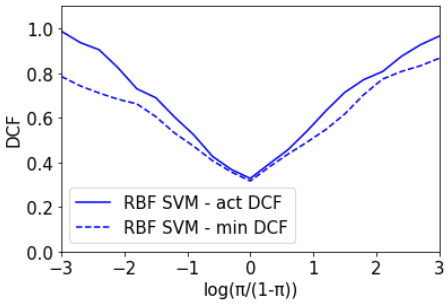


Figure 9: minDCF and actDCF for different working points of the primary system

This is confirmed by a Bayes error plot, which shows the DCFs for different applications: from this it can be noticed that for balanced application priors, scores are well calibrated, while score re-calibration would be needed in case of unbalanced priors.

4. Evaluation

At this point different systems analyzed on the training set have to be compared on the test set in order to verify whether the decisions were effective and the proposed model (SVM with RBF kernel) is the one that performs effectively better.

Finally, actual DCF has to be evaluated on the test set to assess the goodness of score calibration.

4.1. Gaussian classifier

The results are consistent with those obtained on the validation set. The MVG model with tied covariance over raw features provides the best performance ($minDCF(\widetilde{\pi}=0.5)=0.319$) for the primary application, and PCA does not improve the results.

4.2. Logistic regression

4.2.1 Linear logistic regression

First of all, the chosen value of λ has to be verified, ensuring that it provides results close to the optimal ones. The curves for the validation and evaluation set follow the same trend, but the best performances are obtained with $\lambda=0.001$ for the raw features and $\lambda=0.1$ for the gaussianized ones, however the results with the previously selected λ are close.

	$\widetilde{\pi}=0.5$	$\widetilde{\pi}=0.7$	$\widetilde{\pi}=0.1$
Raw features ($\lambda=0.0001$)			
Linear LR ($\pi_T=0.5$)	0.339	0.485	0.701
Linear LR ($\pi_T=0.7$)	0.342	0.474	0.686
Linear LR ($\pi_T=0.1$)	0.323	0.498	0.712
Linear LR	0.334	0.497	0.703
Gaussianized features ($\lambda=0.001$)			
Linear LR ($\pi_T=0.5$)	0.341	0.521	0.693
Linear LR ($\pi_T=0.7$)	0.353	0.525	0.728
Linear LR ($\pi_T=0.1$)	**0.310**	0.475	0.713
Linear LR	0.328	0.497	0.689

Table 8: minDCF for different class priors

In this case, gaussianized features perform better than raw ones for the primary application, in contrast with the results obtained on the validation set. The best results are obtained with $\pi_T=0.1$. The linear regression model provides better performance than the MVG classifier with tied covariance. As concerns the other applications ($\widetilde{\pi}=0.1$ and $\widetilde{\pi}=0.7$) raw features perform slightly better.

4.2.2 Quadratic logistic regression

Again, the value of λ that was previously selected on the validation set has to be verified. The curves for the validation and evaluation set have the same trend. Again the chosen values of λ are not the optimal ones, but the difference of the minDCF is minimal, hence the choice of λ was effective.

	$\widetilde{\pi}=0.5$	$\widetilde{\pi}=0.7$	$\widetilde{\pi}=0.1$
Raw features ($\lambda=0.000001$)			
Quad LR ($\pi_T=0.5$)	0.414	0.617	0.753
Quad LR ($\pi_T=0.7$)	0.393	0.602	0.750
Quad LR ($\pi_T=0.1$)	0.450	0.701	0.754
Quad LR	0.416	0.624	0.739
Gaussianized features ($\lambda=0.001$)			
Quad LR ($\pi_T=0.5$)	0.291	0.366	0.667
Quad LR ($\pi_T=0.7$)	0.288	0.371	0.675
Quad LR ($\pi_T=0.1$)	0.287	0.389	0.688
Quad LR	**0.286**	0.382	0.668

Table 9: minDCF for different class priors

As for the linear analysis, results on the evaluation set are consistent with those on the validation set. Gaussianized features perform significantly better. For the primary application, the model with unbalanced classes provides the best performance, as it happened for the validation set. Training with a different prior ($\pi_T=0.5$) may be helpful for the other two applications.

4.3. SVM

4.3.1 Linear SVM

The hyper-parameter C presents a more regular trend for raw features on the evaluation set compared to the validation one. The lowest value of the minDCF is obtained with $C=0.1$, in contrast with the validation set results ($C=0.001$). Also in this case, the choice of C seems less critical for gaussianized features: the choice ($C=0.1$) was appropriate.

Also in this case, Gaussianization improves significantly the performances.

	$\widetilde{\pi}=0.5$	$\widetilde{\pi}=0.7$	$\widetilde{\pi}=0.1$
Gaussianized features ($C=0.1$)			
Linear SVM ($\pi_T=0.5$)	0.325	0.500	0.694
Linear SVM	**0.309**	0.491	0.705

Table 10: minDCF for different class priors

The best results are obtained without class re-balancing, in contrast with the results on the validation set. This linear model performs again slightly worse than the quadratic logistic regression.

4.3.2 Kernel SVM

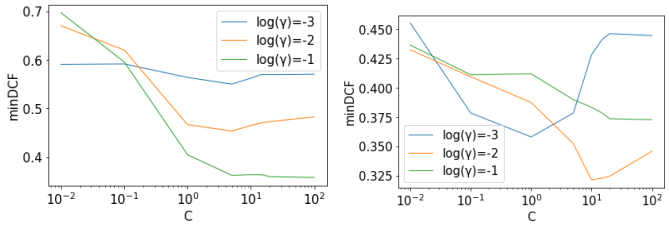
Considering the SVM with quadratic kernel, the values of the hyper-parameters C and c that provides the best performance are significantly different for the primary application ($C=0.01$ and $c=1$). However, the difference between the optimal minDCF and the one obtained with the previously selected values is quite small ($\approx 3\%$).

	$\widetilde{\pi}=0.5$	$\widetilde{\pi}=0.7$	$\widetilde{\pi}=0.1$
Gaussianized features ($C=100, c=0.1$)			
Quad SVM ($\pi_T=0.5$)	**0.300**	0.383	0.689
Quad SVM	0.303	0.410	0.660

Table 11: minDCF for different priors

Class re-balancing slightly improves the performance for the primary application. In contrast with validation set results, quadratic logistic regression provides better outcomes with respect to quadratic kernel SVM.

At this point, the SVM with RBF kernel has to be considered, which represents the selected primary system.



(a) (b)

Figure 10: minDCF ($\widetilde{\pi}=0.5$) for different values of C and γ: (a) raw features, (b) gaussianized features

As for the validation set, gaussianized features achieve the best performances. The choice of the hyper-parameters was however not very effective: indeed, the difference between the optimal result (obtained with $C=1$ and $\gamma=0.001$) and the one with the previously chosen values is about 10%.

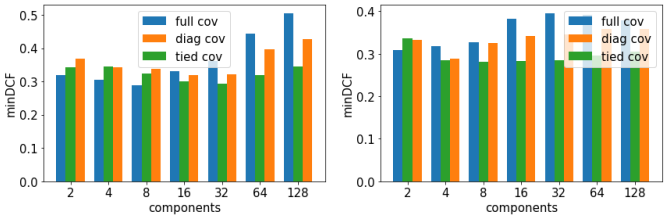
	$\widetilde{\pi}=0.5$	$\widetilde{\pi}=0.7$	$\widetilde{\pi}=0.1$
Gaussianized features ($C=10, \gamma=0.01$)			
RBF SVM ($\pi_T=0.5$)	0.284	0.407	0.640
RBF SVM ($\pi_T=0.7$)	0.311	0.406	0.676
RBF SVM ($\pi_T=0.1$)	**0.278**	0.369	0.679
RBF SVM	0.292	0.382	0.602

Table 12: minDCF for different class priors

Surprisingly, the best results for the primary application are obtained training with $\pi_T=0.1$. Anyway, the results with $\pi_T=0.5$ are close. For the other applications, class re-balancing does not show a precise trend between the used prior and the obtained results. Also for the evaluation set, the SVM with RBF kernel is the model that performs best on this dataset.

4.4. GMM

As far as GMM classifiers are concerned, the choice of the number of components has to be evaluated.



(a) (b)

Figure 11: minDCF ($\widetilde{\pi}=0.5$) for different values of C and γ: (a) raw features, (b) gaussianized features

In this case, gaussianization improves the performances, in contrast with what observed for the validation set. The best model is the tied covariance one with 8 components. The choice relative to the best model and number of components was not effective. However, it does not provide improvements with respect to SVM with RBF kernel, which is still the model that achieves the best results.

4.5. Score quality

Up to now, systems on the evaluation set have been analyzed in terms of minDCFs. But the goodness of the decisions that these systems are actually able to make using the recognizer scores still has to be assessed. For this reason, the actual DCF has to be considered.

The scores provided by the primary system (SVM with RBF kernel) on the validation set were quite calibrated, so class predictions can be made comparing scores with the optimal threshold, which depends solely on the cost of errors and the class priors ($t=-\log\frac{\widetilde{\pi}}{1-\widetilde{\pi}}$).

	$\widetilde{\pi}=0.5$		$\widetilde{\pi}=0.7$		$\widetilde{\pi}=0.1$	
DCF	min	act	min	act	min	act
RBF SVM	0.284	0.303	0.407	0.410	0.640	0.752

Table 13: minDCF and actDCF for different working points of the primary system

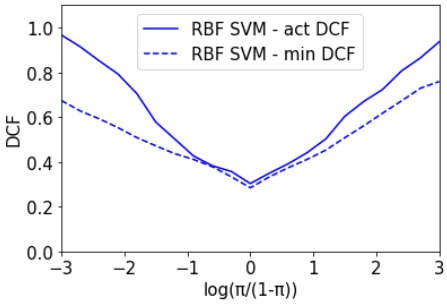


Figure 12: minDCF and actDCF for different working points of the primary system

As expected, the scores are quite calibrated for the primary application, while it would be better to re-calibrate scores if the target application has unbalanced effective prior.

5. Conclusion

Overall, the results obtained on the evaluation set are consistent with those retrieved on the validation set. It is clear that some models provide hyper-parameters that are slightly different but in most cases the difference between the optimal minDCF and the one obtained with the previous values is irrelevant. In the evaluation set, the proposed model SVM with RBF kernel has been confirmed as the best performing model on this dataset so also the choice of the primary system was effective.