Wine Quality Detection

Pietro Macori

s283421

Machine Learning and Pattern Recognition (01URTOV)

October 19, 2022

1 Introduction

This document aims to present and analyze different approaches to classify the quality of Vinho Verde, a Portuguese wine. Specifically, the task is to assign two possible labels to each test sample: good or bad.

For this task, the train and test samples have been retrieved from the UCI repository. Eleven features are used to describe the physical properties of the wines. It must be noticed that in the original dataset, the possible labels range from 0 (bad quality) to 10 (good quality). This would lead to a multi-class classification problem which would be much more complicated than the requested binary task. To solve this issue, the samples have been cast into two groups depending on their labels according to the following principle:

$$binary\_label = \begin{cases} \textbf{bad} & 0 \leq label < 6 \\ \textbf{good} & 6 < label \leq 10 \end{cases}$$

It is important to highlight that samples with quality 6 are removed to simplify the classification task.

In the following chapters, the techniques, the implementation choices, and the results obtained during the development are discussed and analyzed.

2 Data Preprocessing and Visualization

In this chapter, the characteristics of the features of the training samples are reported and analyzed. The training set is composed by 1226 bad-quality samples and 613 good-quality samples, so the classes are not balanced.

As described previously, the wine properties are described by eleven features. As can be observed in 1, the raw features have irregular distributions, and the presence of outliers could lead to sub-optimal results during the classification task. For this reason, a pre-processing step, known as "gaussianization" have been implemented

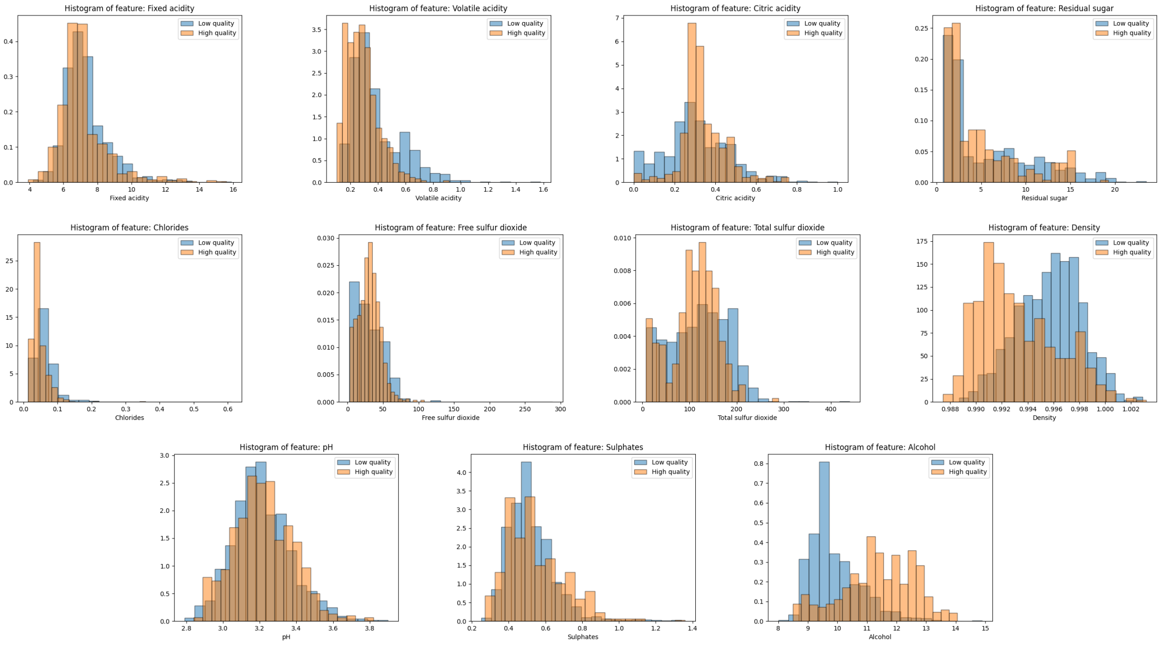


Figure 1: Distribution of raw features

The goal of gaussianizing the raw features is to obtain more regularly distributed data (specifically following a Gaussian distribution) by mapping the features to values with uniform distribution and then transforming them through the inverse of the Gaussian cumulative distribution function. In order to do so, the first task is to compute the rank of the features over the training set using the following formula:

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

Where N is the number of samples of the training set and x_i is the value of the considered feature of the i-th training samples.

Subsequently, the value of the transformed feature is computed as:

$$y = \Phi^{-1}(r(x))$$

The histograms of Figure 2 show the distribution of the dataset features after the gaussianization procedure. As can be seen, the data assumes a much more regular distribution and the previously present outliers have been removed.

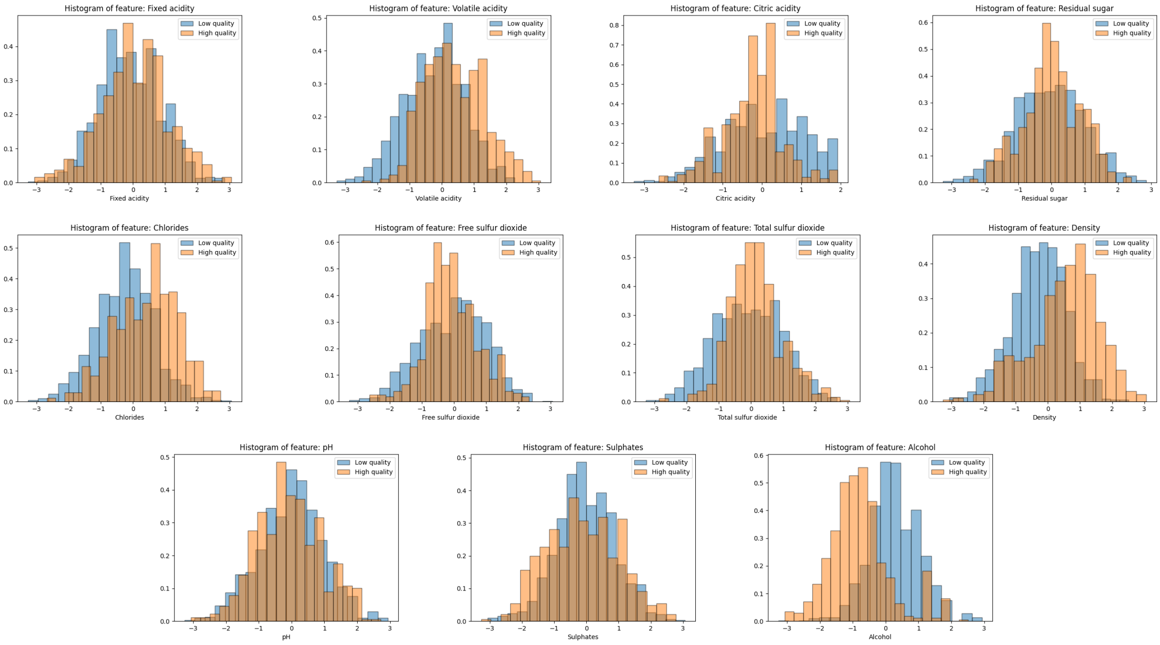
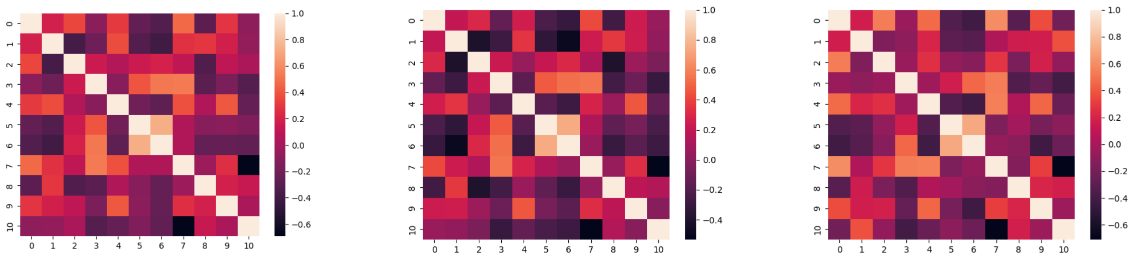


Figure 2: Distribution of features after gaussianization

As a next step, the correlation among features is analyzed. In particular, Figure 3 and Figure 4 show the heat maps of the Pearson correlation coefficient. For completeness, the correlation of the whole dataset and the one for the specific classes is reported for both raw and gaussianized features. It can be observed that some features appear to be correlated (although not strongly). Consequently, the Principal Component Analysis (PCA) can be exploited to retain only uncorrelated features.

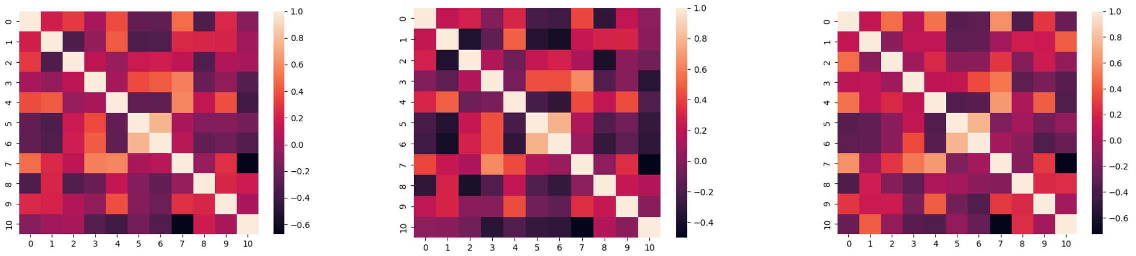


(a) Dataset

(b) Bad Wines

(c) Bad Wines

Figure 3: Heat maps of raw features



(a) Dataset

(b) Bad Wines

(c) Good Wines

Figure 4: Heat maps of gaussianized features

3 Classifiers

3.1 Evaluate the Most Promising Models

Following the dataset's features analysis, different classifiers are applied on the training data, and their performance is assessed using the minimal Detection Cost Function (DCF), the best outcome possible in the case of an ideal threshold choice. Using this approach, it was possible to have an objective way to compare different classifiers. The models are trained and evaluated using 5-fold cross-validation. The application of interest is a uniform prior one with: $\tilde{\pi} = 0.5$, $C_{fp} = 1$, and $C_{fn} = 1$.

3.1.1 Gaussian Models

The first classifiers tested on the training dataset are four different kinds of Gaussian classifiers:

- Full-covariance model
- Naive Bayes model
- Tied-covariance model
- Tied Naive Bayes model

Table 1 reports the minimum Detection Cost Function for all the four cases. The models are tested on raw and Gaussianized features, and dimensionality reduction is applied using PCA. It can be observed that the full-covariance model achieves the best performance. This is probably due to the fact that the dataset is quite large, and as a consequence, it is possible to estimate an entire covariance matrix for each class. There is no need to simplify assumptions like those made by diagonal and tied models. Additionally, the better performance of full-covariance Gaussian models can be justified because they generate quadratic separation surfaces. In contrast, tied-covariance models generate linear separation surfaces, which are less adaptable to the actual data distributions. Diagonal covariance models provide the worst performance, which may indicate that the hypothesis of uncorrelated components does not hold in this case. Moreover, as expected, Gaussianization improves the classification since it shapes the features according to Gaussian distributions. PCA is also helpful in improving the quality of the classifiers, and more specifically, it is very effective in reducing the features to 10. Further reducing the dimensionality (e.g., PCA = 9) does not improve the classification. For this reason, in the following sections, only PCA = 10 will be considered.

Model	minimum DCF	Model	minimum DCF
Raw - no PCA		Gaussianized - PCA = 10	
Full MVG	0.312	Full MVG	0.291
Naive Bayes MVG	0.420	Naive Bayes MVG	0.400
Tied MVG	0.336	Tied MVG	0.351
Naive Tied MVG	0.403	Naive Tied MVG	0.352
Gaussianized - no PCA		Gaussianized - PCA = 9	
Full MVG	0.301	Full MVG	0.303
Naive Bayes MVG	0.439	Naive Bayes MVG	0.405
Tied MVG	0.345	Tied MVG	0.352
Naive Tied MVG	0.442	Naive Tied MVG	0.354

Table 1: Minimum DCF for the four Gaussian classifiers using differently pre-processed data

3.1.2 Logistic Regression

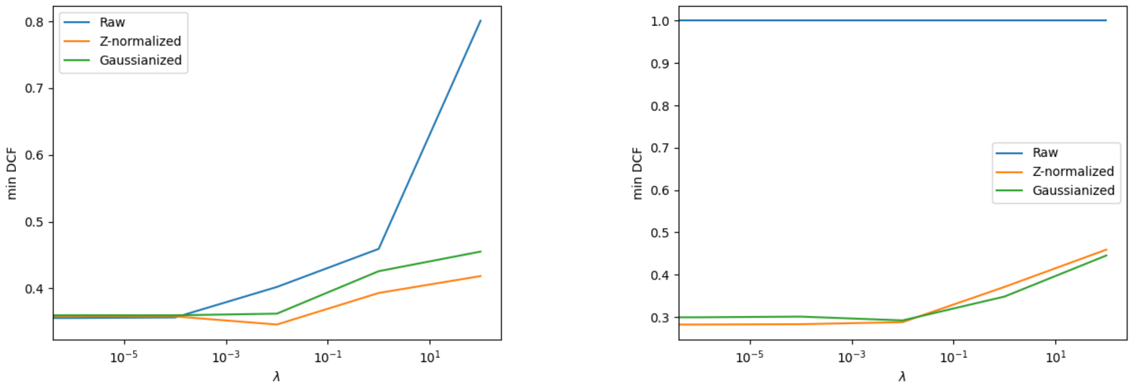
The second model being evaluated is logistic regression. As described in the previous chapter, the classes of the training set are not balanced. To overcome this issue, the cost of the classes is re-balanced using a loss function that considers the different numbers of samples of the two classes. Specifically, the exploited loss function is the following:

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

Where π_T represents the re-balancing term and, for this project, has been set to $\frac{1}{2}$. Instead, the λ value acts as a penalty factor for choosing too large values of w. Its use solves the issue related to the lack of a minimum in the loss function when the classes are linearly separable. To estimate the hyper-parameter λ, different values are tested using cross-validation. Figure 5 shows the relationship between the minimum Detection Cost Function of different regression models and the hyper-parameter λ. As could be expected that the choice of λ has a significant impact on the performance.

Table 1 shows the minimum Detection Cost Function for various type of pre-processed data and the corresponding optimal λ. For this project, both the linear and quadratic logistic regression models are examined and compared. Due to numerical issues (specifically, an overflow), quadratic logistic regression on raw features does not produce useful results. It can be seen that Gaussianization is ineffective in this situation because the models adapt well also to features whose distribution is not Gaussian. To prevent the numerical issues faced with raw data, the Z-normalization of the features is exploited, and it improves the minimum DCF.

Quadratic models perform better than linear ones. This is probably due to their separation



(a) Linear regression

(b) Quadratic regression

Figure 5: Minimum DCF for different values of λ for linear and quadratic regressions

surfaces which are more sophisticated and can better describe the actual distribution of the data without the risk of overfitting. In general, quadratic logistic regression performs similarly (but better) to full-covariance Gaussian models, as they both employ quadratic separation surfaces. On the other hand, linear logistic regression models perform similarly to tied-covariance Gaussian models since they both provide linear separation surfaces.

Eventually, it is possible to notice that PCA does not bring any relevant advantage, although it is not harmful. This appears reasonable since the samples have few dimensions, and the models can benefit from exploiting all of them.

Model	**minimum DCF**	**Model**	**minimum DCF**
Raw - no PCA		Z-normalized - no PCA	
Linear ($\lambda = 10^{-4}$)	0.356	Linear ($\lambda = 10^{-2}$)	0.344
Quadratic ($\lambda = 0$)	1.000	Quadratic ($\lambda = 0$)	0.273
Gaussianized - no PCA		Z-normalized - PCA = 10	
Linear ($\lambda = 10^{-4}$)	0.360	Linear ($\lambda = 10^{-2}$)	0.346
Quadratic ($\lambda = 10^{-2}$)	0.292	Quadratic ($\lambda = 0$)	0.282

Table 2: Minimum DCF for linear and quadratic logistic regression models

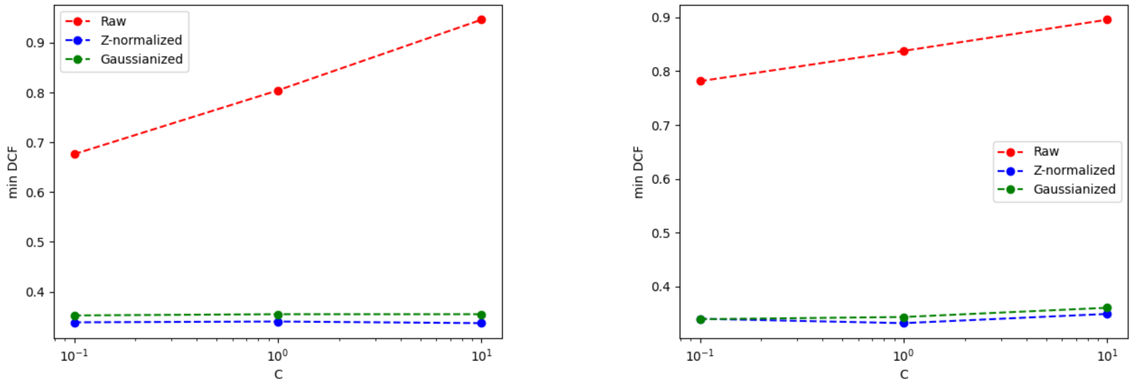
3.1.3 Support Vector Machines

The third analyzed model is Support Vector Machines (SVM). Specifically, the performances of both linear and kernel SVMs are described and analyzed.

As has been explained previously, the classes are unbalanced. For this reason, the performance of the models are evaluated also applying re-balancing. In the context of SVM, the re-balancing is done by modifying the boundaries of the Lagrange multipliers ($0 \leq \alpha_i \leq C$) in the dual formulation of the SVM problem. Specifically, the C values of the two classes are modified in the following way:

$$C_T = C\frac{\pi_T}{\pi_T^{emp}} \qquad C_F = C\frac{1-\pi_T}{\pi_F^{emp}}$$

The terms π_T^{emp} and π_F^{emp} represent the empirical prior values of the two classes. Using the k-fold cross validation methodology, the impact in the performance of different values of hyper-parameter C is measured. As can be seen in Figure 6, C does not impact in a significant way the the minimum DCF value for pre-processed data.



(a) Balanced classes

(b) Unbalanced classes

Figure 6: Minimum DCF for different values of C for linear SVM

Table 3 reports the value of minimum Detection Cost Function for the linear SVM for various pre-processed data. It's evident that using the raw samples leads to a very poor performance. As consequence, a pre-processing phase is required. Looking again to the minimum DCF, it's possible to notice that the gaussianization of the features does not improve the quality of the model. Instead, the Z-normalization is very helpfully in improving the minimum DCF. For this reason only Z-normalized data will be considered in the following considerations. Furthermore, balancing the classes does not take any improvement in the classifier performances. In general, as it was expected, the overall performance of the linear SVM behaves similarly to the previously presented linear models and doesn't provide significant improvements.

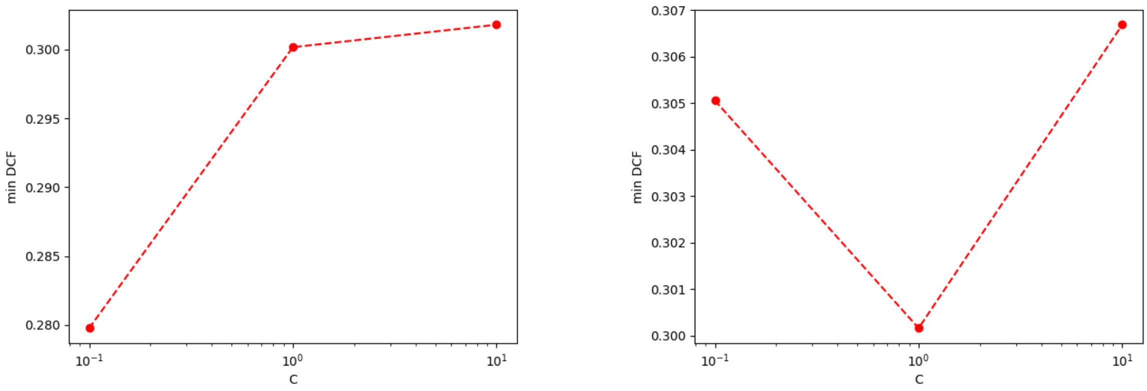
The strength of the dual solution of SVM is that it allows non-linear transformations simply exploiting dot-product in the expanded space. This feature, allows to develop non-linear models by defining some kernel functions. For this project, two kernel function are defined:

- Quadratic kernel
- Gaussian Radial Basis (RBF) kernel

Model	**minimum DCF**
Raw - no PCA	
SVM ($C = 0.1$) - no rebalancing	0.781
SVM ($C = 0.1$) - rebalancing	0.676
Z-normalized - no PCA	
SVM ($C = 1$) - no rebalancing	0.332
SVM ($C = 10$) - rebalancing	0.337
Gaussianized - no PCA	
SVM ($C = 0.1$) - no rebalancing	0.339
SVM ($C = 0.1$) - rebalancing	0.352

Table 3: Minimum DCF for linear SVM models on various pre-processed data

To tune the different hyper-parameters (C for the quadratic kernel, C and γ for the RBF kernel SVM), different values are tested, and the results can be observed in Figure 7 and Figure 8. It can be observed that the results for RBF are quite similar either using class rebalancing or not. The choice of the hyper-parameters appears important in both cases.

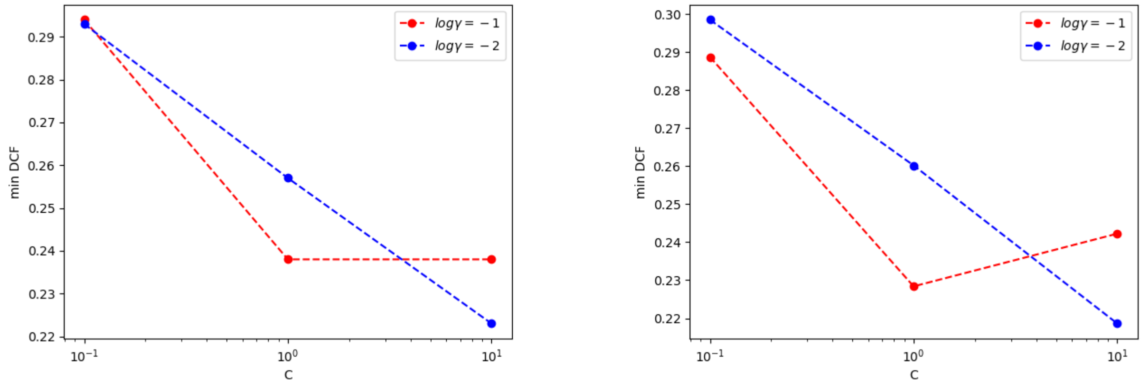


(a) Balanced classes

(b) Unbalanced classes

Figure 7: Minimum DCF for different values of C for quadratic kernel SVM

As can be seen in Table 4, the quadratic kernel SVM performs similarly to the previously shown quadratic models (full-covariance Gaussian and quadratic logistic regression). This is expected behavior since they all support quadratic separation surfaces. However, the RBF kernel SVM performs much better than the previously presented models. This could be explained by the fact that the Radial Basis function supports more complex separation surfaces and, at the same time, avoids overfitting. Eventually, it can be observed that class rebalancing is slightly beneficial but does not significantly improve them.



(a) Balanced classes

(b) Unbalanced classes

Figure 8: Minimum DCF for different values of C for RBF kernel SVM

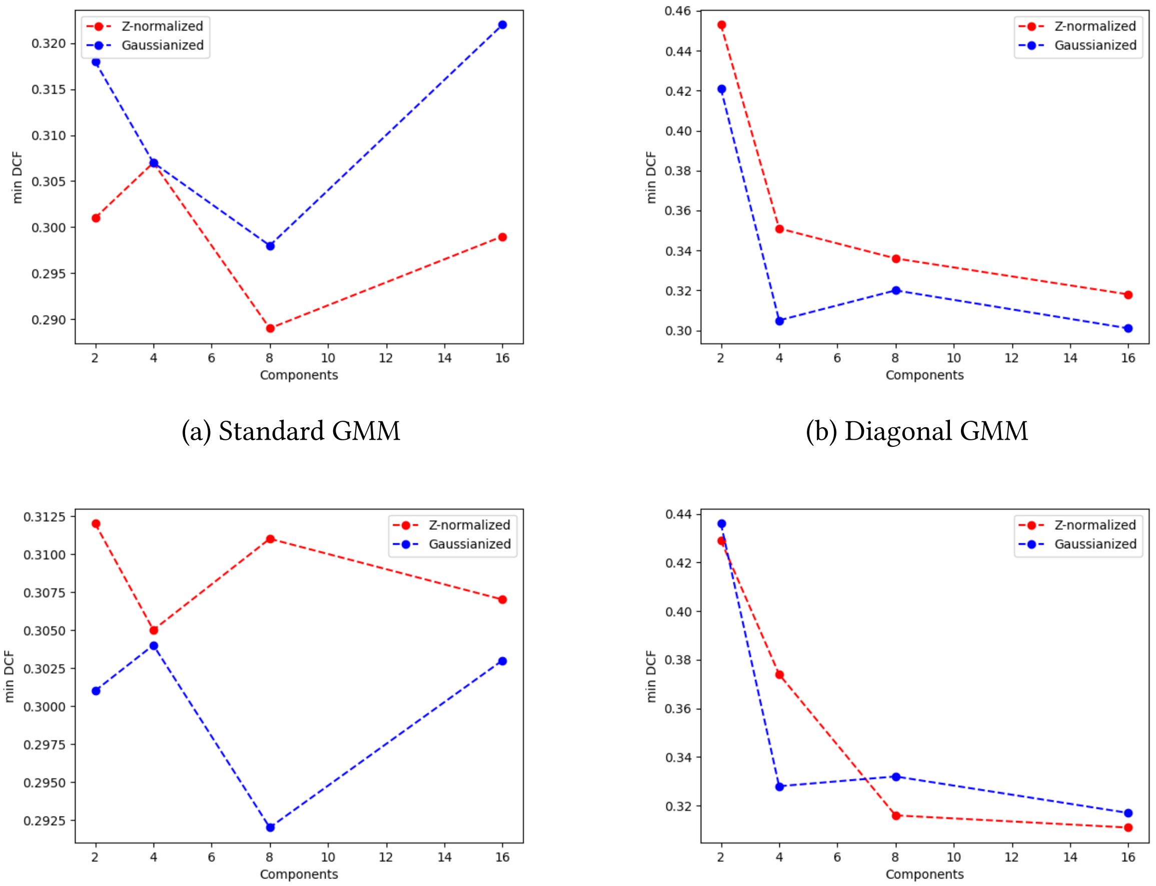
Model	minimum DCF
Z-normalized - no PCA	
Quadratic SVM ($C = 1$) - no rebalancing	0.275
Quadratic SVM ($C = 0.1$) - rebalancing	0.272
RBF SVM ($C = 10$, $log(\gamma) = -2$) - no rebalancing	0.225
RBF SVM ($C = 10$, $log(\gamma) = -2$) - rebalancing	0.223

Table 4: Minimum DCF for quadratic and RBF kernel SVM on various pre-processed data

3.1.4 Gaussian Mixture Model

The last models to be tested are GMMs. The expectation is that GMM models perform better than standard Gaussian ones since they can approximate generic distributions. For completeness, different variants of GMM are analyzed: standard GMM, GMM with diagonal covariance matrices, and tied-covariance GMM (covariance matrix is tied for each class, but different classes can assume different covariance matrices). As usual, the optimal number of components is selected using cross-validation. Figure 9 shows the results for different numbers of components for all the various GMM types. For diagonal models, the increasing number of components seems to regularly decrease the min DCF, whereas, for the other models, there is not a regular and well-defined pattern.

Then, in Table 5 the results for the best choice of components for each model are reported. Given the results obtained using the Gaussian models, the analysis is performed both on Z-normalized features and on Gaussianized features. It can be observed that Gaussianization is not improving the minimum DCF score in this case. The model providing the best performance is the standard GMM, as there are enough data to properly adjust all parameters. However,



(a) Standard GMM

(b) Diagonal GMM

(c) Tied GMM

(d) Tied and diagonal GMM

Figure 9: Minimum DCF for different of components for various types of GMM

also the tied model using Gaussianized features has a very similar performance. Given their greater ability to approximate complicated distributions, models with more components (e.g., 32 or 64) are likely to produce better results, however they were not tested due to their high computation times.

Model	**min DCF**
Z-normalized - no PCA	
GMM (8 comp)	0.289
diag-GMM (16 comp)	0.318
tied-GMM (4 comp)	0.305
tied-diag-GMM (16 comp)	0.311

Model	**min DCF**
Gaussianized - no PCA	
GMM (8 comp)	0.298
diag-GMM (16 comp)	0.301
tied-GMM (4 comp)	0.292
tied-diag-GMM (16 comp)	0.317

Table 5: Minimum DCF of the various types of GMM and the optimal number of components

3.1.5 Conclusion

After having analyzed various classifiers, it can be concluded that the most promising model is the RBF kernel SVM with hyper-parameters $C = 10$, $log(\gamma) = -2$, using rebalancing and Z-normalized features. However, also the quadratic logistic regression ($\lambda = 0$, Z-normalized features, no PCA) and the Gaussian Mixture Model (8 components, Z-normalized features) provide acceptable results. As a consequence, these three models will be taken into consideration in the following sections.

3.2 Selection of the Best Threshold

The next step after selecting the most interesting models based on the minimum Detection Cost Function, is to evaluate their performances using the optimal theoretical threshold. This threshold is computed as:

$$t_{theoretical} = \frac{\tilde{\pi}}{1 - \tilde{\pi}}$$

With $\tilde{\pi} = \frac{1}{2}$. As reported in Table 6, there is a loss of performance for all the considered models, but in general it is not too high. This behavior is quite reasonable since the tested models are not probabilistic, and therefore they do not account for characteristics in the data that can lead to uncalibrated scores.

To overcome this issue, the threshold of the specific application is estimated (t^*). The idea is to use k-fold cross validation. At each iteration of the algorithm, the optimal threshold is selected on the training data and evaluated on the test one. As can be seen from the table below, this approach reduces the DCF for all models, so it's possible to assert that it provide benefits.

Model	**minimum DCF**	**actual DCF($t_{theoretical}$)**	**actual DCF (t^*)**
Quadratric LR	0.272	0.285	0.276
RBF-SVM	0.223	0.232	0.228
GMM (8 comp)	0.289	0.301	0.294

Table 6: Comparison of minimum and actual DCF

3.3 Combining Models

The last step that has to be taken into account is to combine the outputs of different classifiers, so that the final score takes into account models that use different approaches, and as consequence could be better in identifying some specific characteristics of the data. To achieve this, the combined score is computed using the following formula:

$$S = w^T s + c$$

In the expression above s is a vector containing the scores of the different models while w and c are estimated through a logistic regression. Different values of γ of the linear logistic regression are tested.
The models that are combined together are the one that gave better results in the previous section. Specifically, the considered models are :

- Quadratic LR ($\lambda = 0$) on Z-normalized features
- RBF SVM ($C = 10$, $log(\gamma) = -2$) with rebalancing
- GMM with Z-normalized features and 8 components

As it is reported in Table 7, the minimum DCF improves in all cases except for the combination of Support Vector Machines and Gaussian Mixture Model which perform exactly as the SVM alone. For all cases, the actual DCF is slightly worse than the minimum DCF.

Model	**minimum DCF**	**actual DCF(t_{opt})**
SVM + LR	0.214	0.221
SVM + GMM	0.223	0.233
SVM + LR + GMM	0.213	0.225

Table 7: Comparison of minimum and actual DCF for combination of different models

3.4 Conclusion

From the data reported in the previous section it is evident that the performance of the fusion of SVM and LR and the triple fusion of SVM, GMM and LR are very similar results in terms of min DCF. The final model suggested for the task is the most complex of the two because it is reasonable to assume that a more complex model is able to better capture the various characteristics of the dataset. For his reason the choosen classifier is the combination of RBF kernel SVM (C = 10, $log(\gamma) = -2$, rebalancing, Z-normalized features), quadratic logistic regression (γ = 0, Z-normalized features) and GMM (8 components, Z-normalized features), whose scores are combined through a linear model estimated usign a linear logistic regression.

4 Evaluation

After analyzing and justifying the performances of previously presented models on the training set and choosing the most convincing ones, the effectiveness of the decisions is evaluated by computing the performance of the models on the test samples. The chosen hyper-parameters are the ones performing better using the training set. Results are given in the form of minimum Detection Cost Function.

4.1 Single Models

Table 8 shows the results for Gaussian models, whose performances in specific cases present some differences with respect to the model evaluated using the training set. The best model is the tied-diagonal MVG model, which was not so effective when it was evaluated on the training set. The full-covariance model, which previously showed outstanding results in terms of minimum DCF, now is no more the most performing strategy. On the other hand, PCA still provides a positive contribution. As analyzed in the previous chapter, since Gaussian classifiers are considered, the gaussianization of the features greatly improves the models' performances.

Model	**minimum DCF**	**Model**	**minimum DCF**
Raw - no PCA		Gaussianized - PCA = 10	
Full MVG	0.337	Full MVG	0.320
Naive Bayes MVG	0.367	Naive Bayes MVG	0.332
Tied MVG	0.315	Tied MVG	0.313
Naive Tied MVG	0.369	Naive Tied MVG	0.310
Gaussianized - no PCA		Gaussianized - PCA = 9	
Full MVG	0.336	Full MVG	0.326
Naive Bayes MVG	0.377	Naive Bayes MVG	0.338
Tied MVG	0.320	Tied MVG	0.322
Naive Tied MVG	0.379	Naive Tied MVG	0.317

Table 8: Minimum DCF of test set for Gaussian models

Table 9 shows the results for logistic regression models that are similar to the ones on the training set. PCA seems to be more helpful than expected, but, as can be noticed from the slight reduction of the DCF, the improvement it brings is not fundamental.

As was expected from the previous evaluations, quadratic logistic regression performs much better than linear logistic regression. In general, the test set results are comparable with the ones obtained in the previous chapter.

Model	**minimum DCF**
Raw - no PCA	
Linear ($\lambda = 10^{-4}$)	0.338
Quadratic ($\lambda = 0$)	1.000
Gaussianized - no PCA	
Linear ($\lambda = 10^{-4}$)	0.341
Quadratic ($\lambda = 10^{-2}$)	0.284

Model	**minimum DCF**
Z-normalized - no PCA	
Linear ($\lambda = 10^{-2}$)	0.331
Quadratic ($\lambda = 0$)	0.265
Z-normalized - PCA = 10	
Linear ($\lambda = 10^{-2}$)	0.329
Quadratic ($\lambda = 0$)	0.260

Table 9: Minimum DCF of test set for linear and quadratic logistic regression models

Moving to the results obtained using Support Vector Machines models, Table 10 reports the values of the minimum DCF for the linear SVM. In this case, the results obtained by applying the gaussianization pre-processing to the features are better than the ones obtained after Z-normalization. As expected, the raw samples have a much worst performance. Similarly to the behavior obtained using the training set, the linear SVM has comparable results to other linear classifiers.

Regarding kernel SVM models, Table 11 shows the performances of quadratic and RBF kernel SVM. Again the Gaussian Radial Basis function has a lower minimum DCF since it allows more complex separation surfaces. As expected, the quadratic kernel SVM performs similarly to other quadratic classifiers. Rebalancing does not provide significant improvements in the performances of the models. In general, the obtained results are fully comparable to the one presented in Chapter 3.

Model	**minimum DCF**
Raw - no PCA	
SVM ($C = 0.1$) - no rebalancing	0.475
SVM ($C = 0.1$) - rebalancing	0.726
Z-normalized - no PCA	
SVM ($C = 1$) - no rebalancing	0.319
SVM ($C = 0.1$) - rebalancing	0.331
Gaussianized - no PCA	
SVM ($C = 0.1$) - no rebalancing	0.309
SVM ($C = 0.1$) - rebalancing	0.323

Table 10: Minimum DCF of test set for linear SVM models on various pre-processed data

Model	minimum DCF
Z-normalized - no PCA	
Quadratic SVM ($C = 1$) - no rebalancing	0.267
Quadratic SVM ($C = 0.1$) - rebalancing	0.278
RBF SVM ($C = 10$, $log(\gamma) = -2$) - no rebalancing	0.258
RBF SVM ($C = 10$, $log(\gamma) = -2$) - rebalancing	0.260

Table 11: Minimum DCF of test set for quadratic and RBF kernel SVM

Eventually, the Gaussian Mixture Model is evaluated. Similarly to the Gaussian model case, the tied covariance model with four components is the one performing better for training data. Moreover, using GMM, the gaussianized feature seems to improve the classification task. Table 12 reports the values of minimum DCF for the various types of GMM.

Model	min DCF
Z-normalized - no PCA	
GMM (8 comp)	0.306
diag-GMM (16 comp)	0.329
tied-GMM (4 comp)	0.304
tied-diag-GMM (16 comp)	0.325

Model	min DCF
Gaussianized - no PCA	
GMM (8 comp)	0.328
diag-GMM (16 comp)	0.320
tied-GMM (4 comp)	0.284
tied-diag-GMM (16 comp)	0.305

Table 12: Minimum DCF of the various types of GMM and the optimal number of components

4.2 Score Calibration

As has been done before, optimal threshold selection is evaluated. Table 13 shows the results on the three selected models. The optimal threshold (t*) is the best threshold for the complete training set. It can be observed that the loss of performance due to uncalibrated scores is relevant, especially for the GMM. Optimal threshold selection is helpful for the GMM classifier, but it lead to worst results for the other two cases.

Model	minimum DCF	actual DCF($t_{theoretical}$)	actual DCF (t^*)
Quadratric LR	0.265	0.286	0.309
RBF-SVM	0.260	0.278	0.283
GMM (8 comp)	0.306	0.340	0.321

Table 13: Comparison of minimum and actual DCF

4.3 Model Combination

Eventually, Table 14 reports the performances of the different combined models. Actual DCF is obtained by using the theoretical threshold, since model fusions provides calibrated scores. The values of γ for the logistic regression model performing the fusion are the best ones tested during model tuning. It can be observed that all the fusions are, in general, better than to the single models, both in terms of minimum DCF and actual DCF. Indeed, the loss of performance is lower than the one observed for single models, meaning that model fusion provides also score calibration.

In general, the evaluation dataset has similar characteristics with respect to the training dataset, as the results obtained by the two cases are typically compatible. Nevertheless, the final performance of the fusion is worse than the one achieved on the training set, and some discrepancies were observed also in the single models.

Model	**minimum DCF**	**actual DCF(t_{opt})**
SVM + LR	0.253	0.262
SVM + GMM	0.259	0.262
SVM + LR + GMM	0.251	0.259

Table 14: Comparison of minimum and actual DCF for combination of different models

4.4 Other Applications

As last step, it is interesting to analyze the performance of the best models for different applications. Figure 10 shows the ROC plots for the three single models, the three model fusions, and a comparison between single models and the fusion of the three models. It can be observed that the GMM has lower performances than the other two single models, whereas the three fusions have very similar performances across all operating points. Moreover, the fusion achieves better results than single models for the operating points of interest in the task, but there are operating points for which the single logistic regression performs better than the fusion.

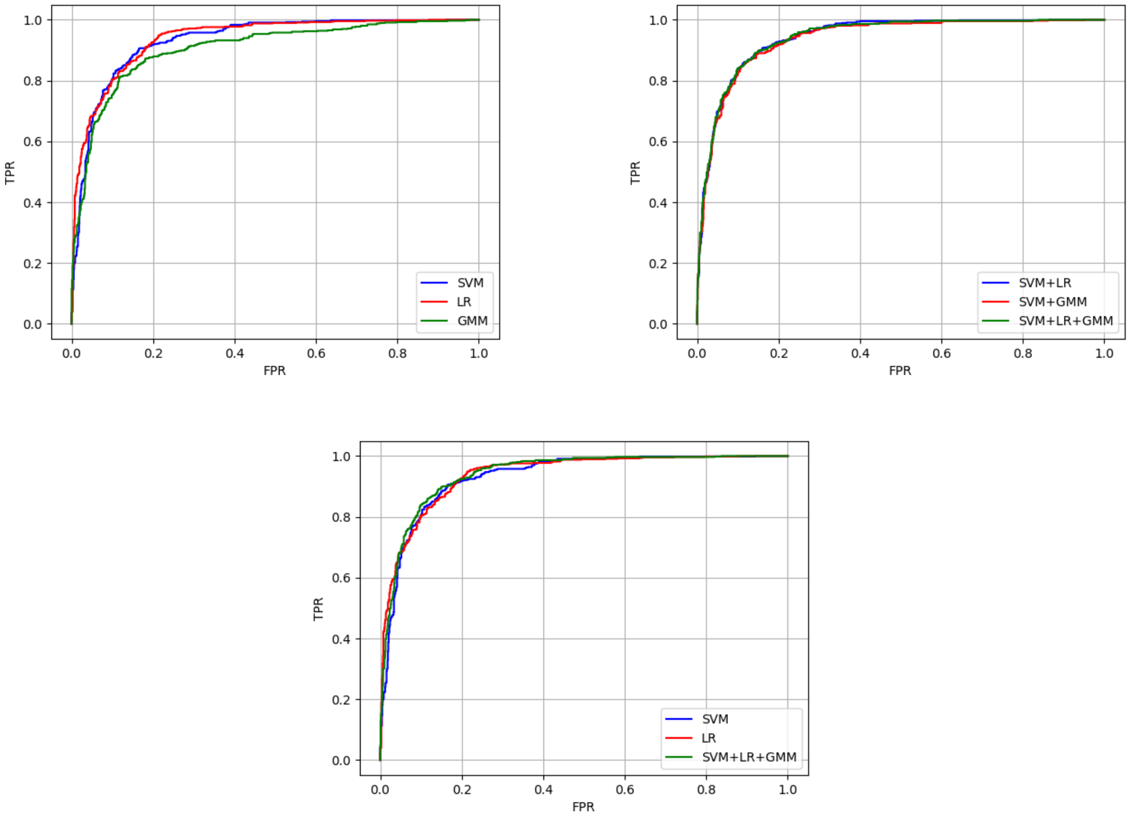


Figure 10: ROC plots of various models

Eventually, Figure 11 reports the Bayes error plots for the various models, reporting the min DCF and actual DCF for different values of the prior log-odds, which corresponds to the different applications.

It is crucial to notice that among the single models, the GMM performs worse than the other two for many applications, and the performance loss due to uncalibrated scores is quite relevant for many applications (especially GMM and logistic regression). The three model combinations have similar performances across all applications, and their scores are well-calibrated. The fusion of three models provides better performance with respect to single models for the application of interest in the task. However, in some applications, the logistic regression model is better than the fusion.

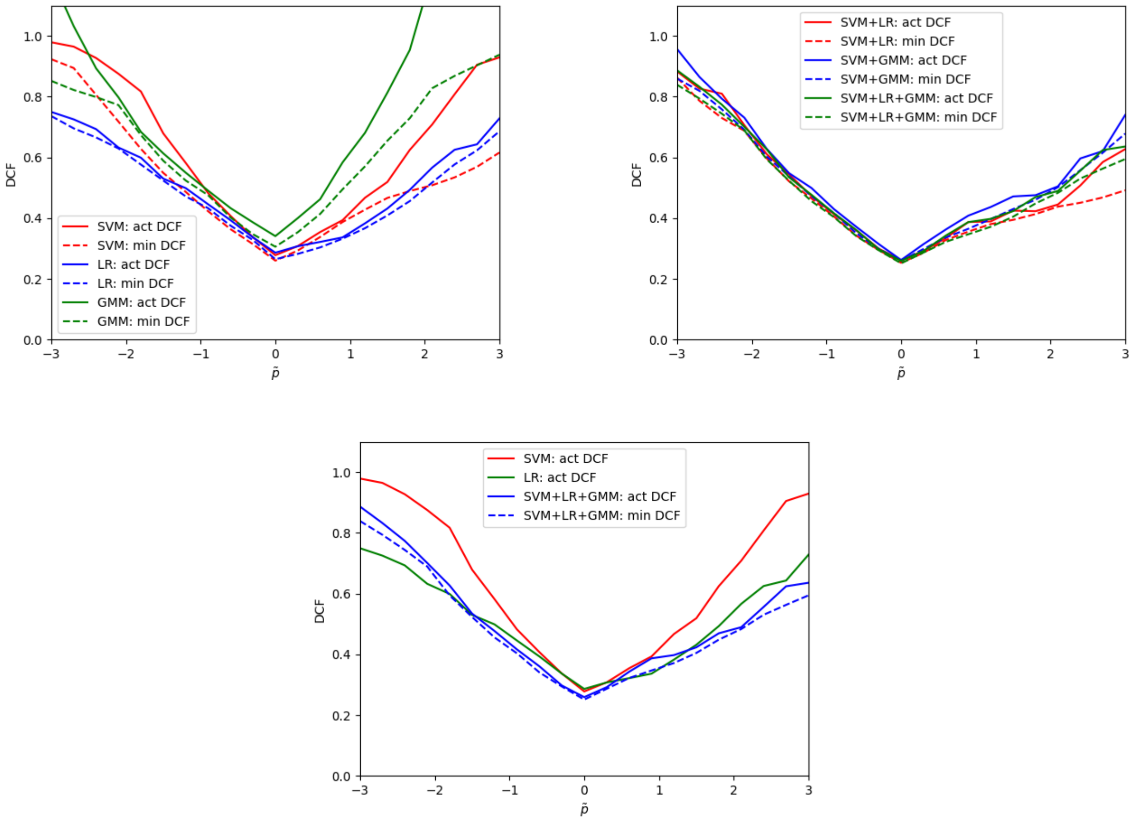


Figure 11: Bayes error plots