Wine Quality Detection
Torta Francesco – S287899
Voto Giorgio - S288179

Model for the Wine quality detection problem

Torta Francesco, Voto Giorgio
Politecnico di Torino, Italy

1. Introduction

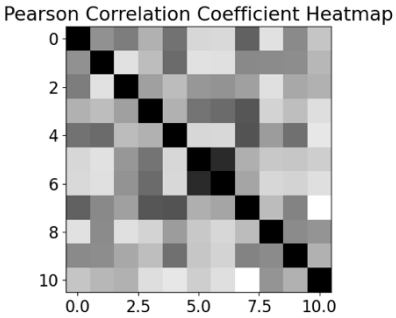
In this report we will discuss different models able to determine, with a good degree of accuracy, the quality of wine. The original dataset is taken from UCI repository but has been modified to transform the task into a binary classification problem, with the objective to label the wine as good or bad in terms of quality.

2. Feature Analysis

In the dataset there are 11 features:

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

A first analysis of the characteristics shows that in many cases, the features have an uneven distribution. For this reason, we will also analyze a dataset pre-processed by Gaussianization, since, especially for models based on Gaussian, we would obtain suboptimal results. The heat map that displays the Pearson Correlation Coefficient shows us that some characteristics are related to each other (e.g., 5-6).



While this is true, the correlation is not strong enough to use a feature size reduction technique such as PCA in our models. This technique will be used only in Gaussian-Based models for completeness of discussion of the problem. In general, we will only use K-Fold cross validation with K = 5 to obtain more robust results.

We will analyze the models with 3 different applications:

1. $(\tilde{\pi}, C_{fp}, C_{fn}) = (0.5, 1, 1)$
2. $(\tilde{\pi}, C_{fp}, C_{fn}) = (0.1, 1, 1)$
3. $(\tilde{\pi}, C_{fp}, C_{fn}) = (0.9, 1, 1)$

The main application will be the first. In the other two cases we will have unbalanced tasks that prioritize one of the two class.

3. Multivariate Gaussian Classifiers

The first models analyzed are the generative Gaussian classifiers:
-MVG (Full-Covariance)
-NBG (Diagonal-Covariance)
-TCG (Tied Full-Covariance)
-TCNB (Tied Diagonal-Covariance).

We start by doing a selection of the best models in terms of minimum DCF, which is the cost we would pay if we made optimal decision on the test set (validation set) using recognizer scores.

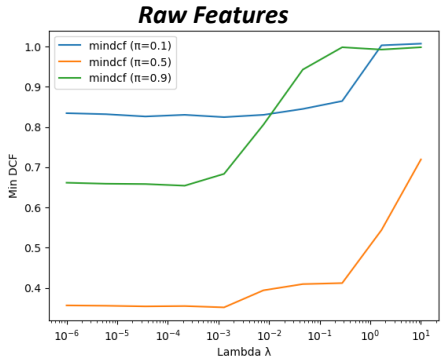
	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Raw Features			
Full-Cov	**0.312**	0.778	0.842
Diag-Cov	0.420	0.845	0.921
Tied Full-Cov	**0.333**	0.812	0.748
Tied Diag-Cov	0.403	0.866	0.932
Gauss Features			
Full-Cov	**0.299**	0.811	0.789
Diag-Cov	0.446	0.820	0.882
Tied Full-Cov	0.347	0.788	0.848
Tied Diag-Cov	0.452	0.866	0.930
Gauss Features – PCA = 10			
Full-Cov	0.729	1.007	0.998
Tied-Full-Cov	0.788	1.005	0.998

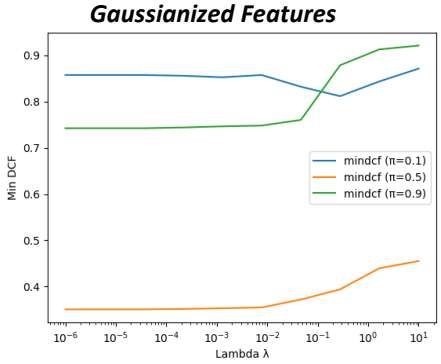
MVG Raw, TCG Raw and MVG Gaussianized are the best 3 models. Diagonal models are in general worse. PCA worsens the minDCF in all cases, in fact from the Pearson Correlation Coefficient Heatmap we can notice how no pair of features are highly correlated between each other. Quite all models are useless for imbalanced tasks.

4. Linear Logistic Regression

We start considering the linear Logistic Regression model. This is a discriminative approach for classification. Its log-likelihood ratio is a linear function like the Tied Covariance Gaussian one, so we expect to have similar results (even though the logistic regression model does not make any assumption on the data distributions). We will use the model with class balancing.

Given $\pi_T = \frac{1}{2}$, we start comparing our three applications by the minDCF metrics, changing the value of the parameter λ.





The regularization of $\|w\|$ improves significantly the accuracy. Our best results are obtained with $\lambda = 10^{-5}$. Even though the logistic regression does not make any assumption on the distribution of the data, the gaussianized features have slightly better results than raw features.
We thus select $\lambda = 10^{-5}$

For different priors π_T, we compute different minDCF for both raw and gaussianized features.

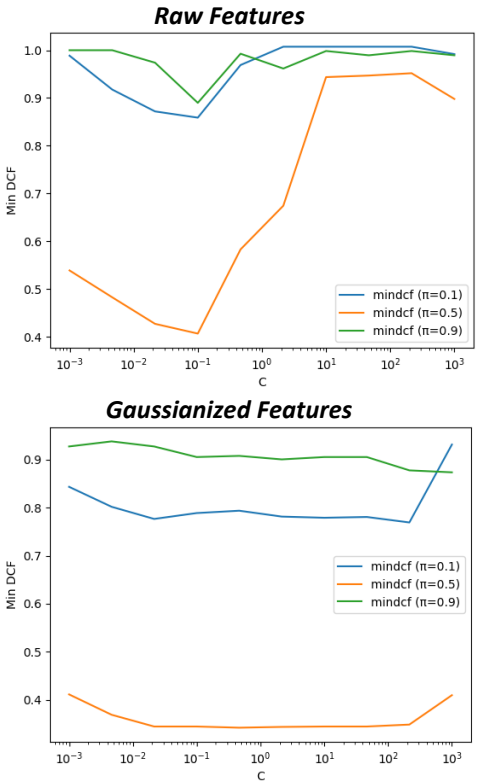
	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Raw Features			
$(\lambda = 10^{-5}, \pi_T = 0.5)$	0.355	0.830	0.660
$(\lambda = 10^{-5}, \pi_T = 0.1)$	**0.338**	**0.821**	0.710
$(\lambda = 10^{-5}, \pi_T = 0.9)$	0.369	0.858	**0.643**
Gauss Features			
$(\lambda = 10^{-5}, \pi_T = 0.5)$	0.351	0.857	0.742
$(\lambda = 10^{-5}, \pi_T = 0.1)$	**0.338**	**0.788**	0.899
$(\lambda = 10^{-5}, \pi_t = 0.9)$	0.374	0.894	**0.687**

Best performances for our main application are obtained by setting π_T to 0.1. Gaussianized features have slightly better results. As we can

see, the values are comparable to the Tied Covariance Gaussian.

5. Linear SVM

To complete the linear models, we will consider the linear SVM. We will use both the unbalanced model and the model with class balancing, which consists in defining different constraint on α_i depending on whether the class c_i is 1 or 0.
The images below show for different values of the parameter C, the minDCF for a linear SVM without class balancing, both for raw and gaussianized features.



A bit surprisingly, the gaussianization of the features improves dramatically the performance of the model.
For the gaussianized feature, the choice of C does not seem to be relevant between 0.1 and 10; for this reason we will select C=1.

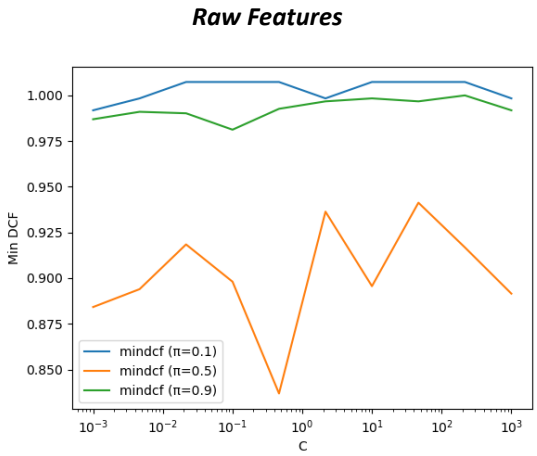
Since linear SVM apparently does not outperform other linear models in accuracy, we have decided to calculate only the minDCF for the gaussianized features with π_T = 0.5 and without class balancing.

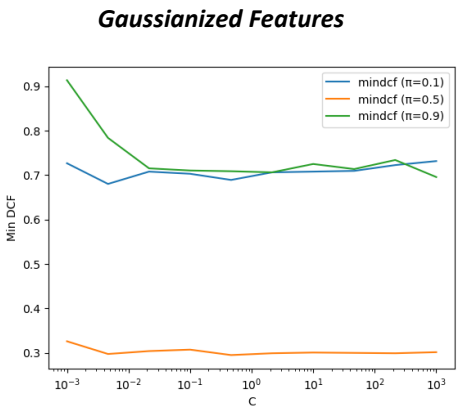
	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Gauss Features			
$(C = 1, \pi_T = 0.5)$	0.346	0.832	**0.798**
$(C = 1)$	**0.343**	**0.780**	0.903

As expected, the results obtained are very similar to those of the logistic regression.

6. Quadratic Kernel SVM

Since the results obtained from the linear models are almost identical, we will focus on the quadratic ones, since they seem to have better performances, as we could see by the MVG Full-Covariance model. As for the linear SVM, we will use both with and without class balancing.
In the plots below, we can see how the parameter C affects the performances of the model. We will focus on our main application (orange line)





Again, gaussianization greatly improves the performance of the classifier. Subsequent

analyses will be carried out exclusively on gaussianized data.
In this case, the choice of C does not affect the results a lot, so we select $C = 1$

The grid below shows the minDCF for different model given the hyperparameters selected.

	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.9$
Gauss Features			
No class-balancing	**0.299**	**0.704**	0.708
$\pi_T = 0.5$	0.307	0.767	0.696
$\pi_T = 0.1$	0.337	0.741	0.795
$\pi_T = 0.9$	0.329	0.865	**0.630**

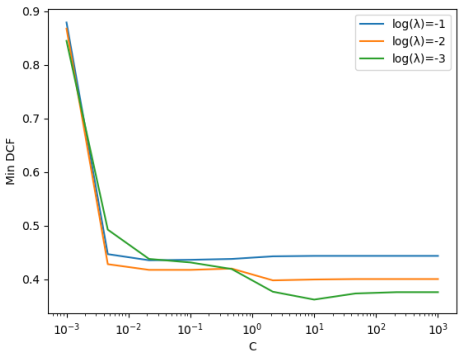
Changing the prior π_T does not lead to huge differences in results. The best model is the one without class balancing.

As expected, the results are very similar to the one obtained by the Full Covariance model with gaussianized features.

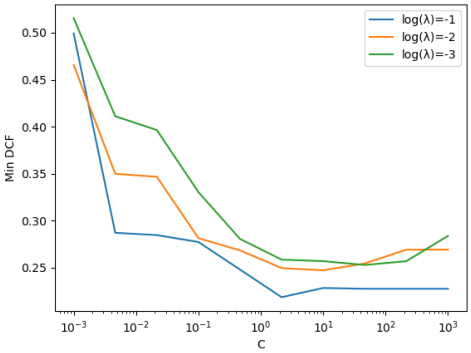
7. Kernel SVM – RBF

Since we have noticed that non-linear models have good performance on this dataset, let's proceed with the analysis of the SVM with RBF kernel.
The RBF SVM needs two parameters, which must be jointly optimized to obtain the best performances of the model.
The grid below shows which are the best λ e C values for our main application.

Raw Features



Gaussianized Features



Also, in this case the gaussianized features considerably increase the efficiency of the model.
Both the choice of lambda and C greatly influence the results.
For the Gaussianized dataset, the best results are obtained with:

$$\log(\lambda) = -1, C = 2$$

For the raw dataset, the selected hyperparameters are:

$$\log(\lambda) = -3\,, C = 10$$

The following results show, for the hyperparameter selected, the minDCF obtained for the gaussianized and raw case.

	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$
Gauss Features			
No class-balancing	**0.508**	**0.218**	0.669
$\pi_T = 0.1$	0.570	0.247	0.764
$\pi_T = 0.5$	0.530	0.223	0.653
$\pi_T = 0.9$	0.666	0.233	**0.567**
Raw Features			
No class-balancing	0.741	**0.362**	0.881
$\pi_T = 0.1$	0.786	0.400	0.927
$\pi_T = 0.5$	0.741	0.370	0.885
$\pi_T = 0.9$	**0.717**	0.384	**0.873**

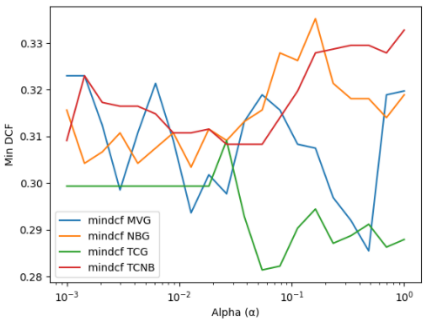
For now, our best results are obtained with an unbalanced SVM classifier using RBF Kernel and the gaussianized features.

8. Gaussian Mixture Models

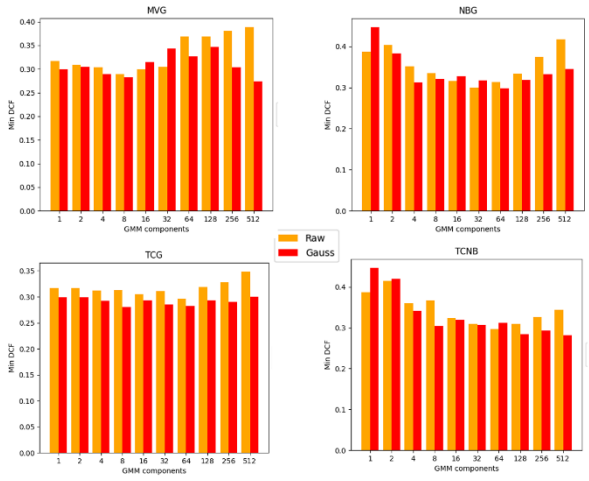
Gaussian Mixture Models are generative models that can approximate generic distributions. We expect in fact to obtain better results than Gaussian Models. We analyze GMM with diagonal and full covariance, with

and without tied covariance (tying takes place at (sub-)class level).

For all models the threshold for the stopping criteria of the EM algorithm is set at 1e-6, and Ψ=0.01 to avoid degenerate cases. Also, hyperparameter α (used to split GMMs in LBG algorithm) is set to 0.1. The plot below shows that the minimum DCF doesn't change too much for low values of α (<0.05). [$\tilde{\pi}$=0.5]



We now turn our attention to find the best number of Gaussian components (G) for all 4 Gaussian models. We try to plot the minimum DCF with different values of G (from 1 to 512) and with raw and gaussianized data, obtained from K-Fold Protocol on validation set.



Diagonal covariance models perform worse, except for middle values of G. Tied models perform better than non-tied ones, especially on raw data. Gaussianization seems effective most of the cases. The best models are the full-covariance 512G and tied full-covariance 8G (both gaussianized).
We will analyze those 2 cases in the next paragraphs.

9. Recalibration of scores

Up to now we have considered the minimum DCF as the only metric, which corresponds to the cost we would pay using a threshold that optimizes decisions based on the scores and labels associated with them. However, it is necessary to choose an optimal threshold for which to classify the scores without knowing the labels a priori. So now we will analyzing the Actual DCF

If the scores are well calibrated, then the threshold corresponding to the optimal Bayes risk is $t = -\log\left(\frac{\tilde{\pi}}{1-\tilde{\pi}}\right)$. We can therefore proceed with the recalibration of the scores of our best models, considering a function f (s), such as to transform the scores *s* into calibrated scores s_{cal}. Assuming that this function is linear $f(s) = \alpha s + \beta$, it is possible to use a prior-weighted Logistic Regression to determine the parameters α and β by training on the scores determined by the models.

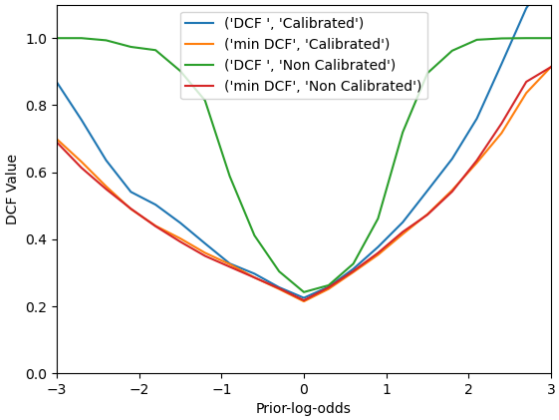
1. SVM RBF

In the grid below, we can see that the model does not have well calibrated scores. This is especially true, for the unbalanced applications.

	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$
minDCF	0.508	0.218	0.669
actDCF	0.985	0.242	0.996

Using the logistic regression trained over the scores, we should achieve better results.

	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$
Actual DCF			
(Uncalibrated)	0.985	0.242	0.996
$\tilde{\pi} = 0.1$	0.566	0.230	0.808
$\tilde{\pi} = 0.5$	**0.564**	0.225	0.827
$\tilde{\pi} = 0.9$	0.644	**0.219**	**0.823**



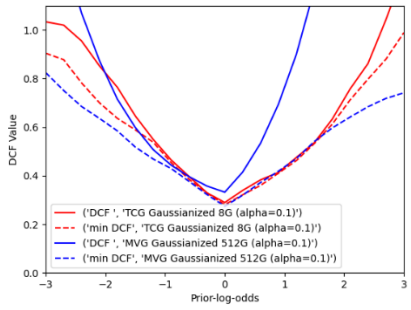
The actual DCF is now similar to the minimum DCF, so we should achieve better results for unseen data.

2. GMM

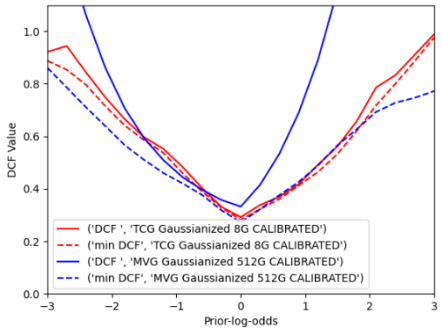
The following table shows the minDCF and (actual)DCF for different applications ($\tilde{\pi}$ = 0.5, 0.1, 0.9) trained with our best models chosen early. The scores are uncalibrated.

	minDCF	*actDCF*
$\tilde{\pi} = 0.5$		
Full-Cov Gau 512G	0.274	0.332
Tied Full-Cov Gau 8G	0.280	0.289
$\tilde{\pi} = 0.1$		
Full-Cov Gau 512G	0.650	0.937
Tied Full-Cov Gau 8G	0.725	0.895
$\tilde{\pi} = 0.9$		
Full-Cov Gau 512G	0.657	2.254
Tied Full-Cov Gau 8G	0.751	0.774

The calibration is quite poor, especially for $\tilde{\pi}$=0.9. This is confirmed from the Bayes error plot which shows different application priors:



If we calibrate the scores with the logistic regression approach, we can obtain better results, although the MVG model with 512 G doesn't calibrate well enough for higher $\tilde{\pi}$ (>0.7). (Note: the choice of $\tilde{\pi}$ in the logistic regression approach doesn't influence too much the calibration).
This is the Bayes Plot for the two models calibrated with $\tilde{\pi}$=0.5:



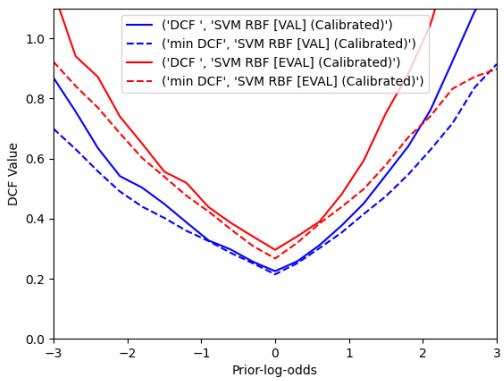
10. Evaluation

We have seen that our two best classifiers are the RBF SVM and the GMMs with Full Covariance (8 clusters) and Tied Covariance (512 clusters). We need to check the quality of our models on the unseen data (the evaluation set).
We will consider only our best models and we will compare their performances with the results obtained on the validation set.

1. SVM RBF

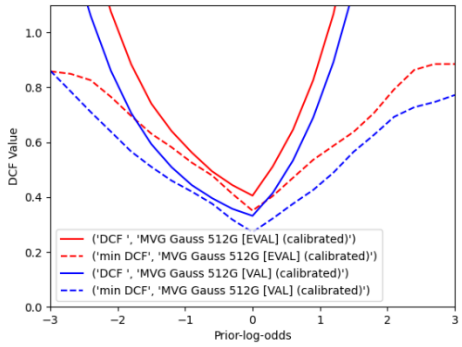
As we can see, for the main application, the results are slightly worse than those obtained on the validation set.

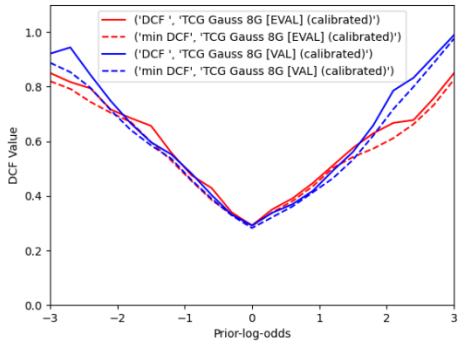
	$\tilde{\pi} = 0.1$	$\tilde{\pi} = 0.5$	$\tilde{\pi} = 0.9$
Validation Set			
minDCF	0.508	0.218	0.669
actDCF (calibrated)	0.564	0.225	0.827
Evaluation Set			
minDCF	0.715	0.267	0.768
actDCF (calibrated)	0.783	0.296	1.117



2. GMM

We evaluate how the two best GMM models perform on unseen data (evaluation set). The plots below show a comparison of the K-Fold protocol results on validation set and the results on evaluation set.





We can observe that the results on validation and evaluation sets are consistent for the TCG 8G model (lower gap). The MVG 512G model has a higher gap, so performs worse on new data, probably due to the overfitting of the 512 components.
We can conclude that the best model among the two is indeed the Tied Covariance with 8 Gaussian components and features' gaussianization, because it performs better in more applications ($\tilde{\pi}$) and computes better decisions on new data.

3. Final Considerations

In conclusion our best models are the SVM RBF and GMM with Tied Covariance and 8 Gaussians, both with gaussianized features. Both perform well on unseen data. The Bayes plot shows that the SVM model can achieve better results on our main application ($\tilde{\pi}$ =0.5). The GMM model, on the other hand is more versatile and performs better on imbalanced applications.