Cazzola Michele s323270

# MACHINE LEARNING AND PATTERN RECOGNITION

Project task

## Laboratory 2 – Loading and visualization

1. For each feature, both classes exhibit a unimodal distribution with Normal shape, and they overlap in the central part of their domain:
   * the mean values are the same (approximately 0) for both features and their respective classes.
   * the variances are different:
     + approximately 0.6 for the fake class of the first feature and the genuine class of the second feature.
     + approximately 1.4 for the genuine class of the first feature and the fake class of the second feature.

We observe that, for each feature, the class with the lower variance exhibits the highest modal frequency (peak value).

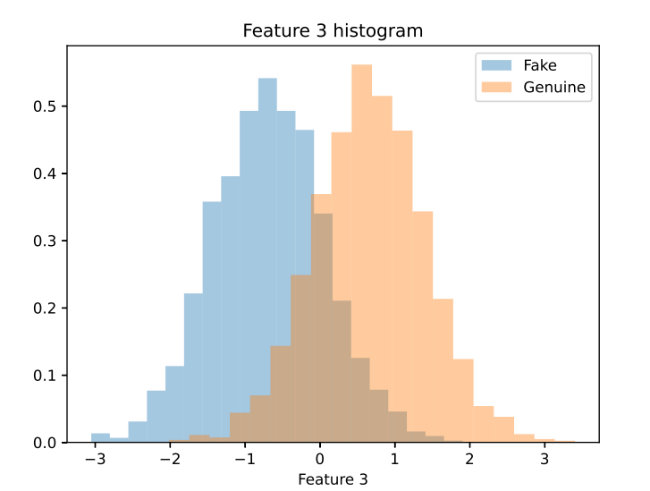
Immagine che contiene diagramma, schermata, testo, Diagramma

Descrizione generata automaticamenteImmagine che contiene diagramma, schermata

Descrizione generata automaticamente

1. For each feature, both classes demonstrate a unimodal distribution with Normal shape, and they overlap on their respective sides; for each pair of classes within each feature:
   * the mean values are opposite but nearly equal in magnitude (between 0.6 and 0.7).
   * the variances are nearly equal (between 0.5 and 0.6).

We observe that, for each feature, the classes display the similar modal frequencies.

Immagine che contiene diagramma, testo, schermata, Diagramma

Descrizione generata automaticamente

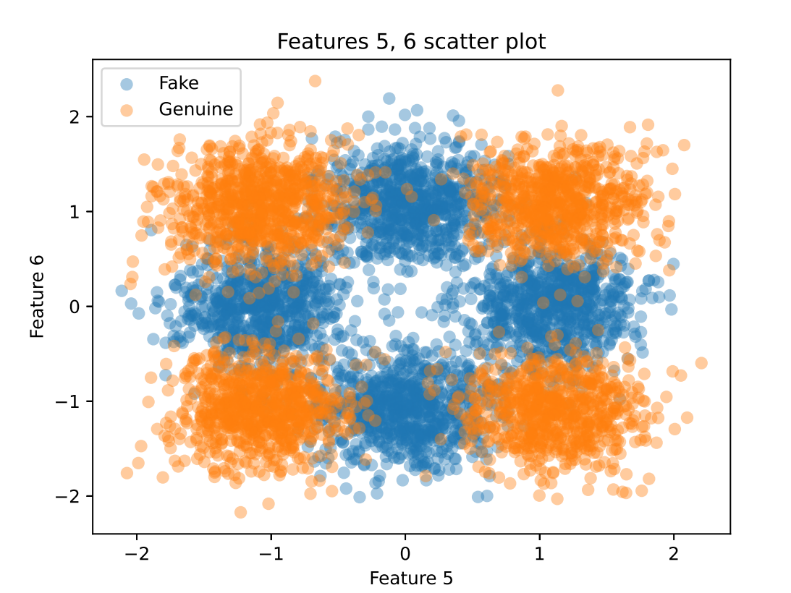
1. For each feature, the fake class displays a unimodal distribution, while the genuine class exhibits a bimodal distribution:
   * for the fake class, the modal values are opposite.
   * for the genuine class, the modal value is approximately zero.

We observe that they overlap around the modal values of the fake class distribution, while the overlapping is minimal in the central part of the domain.

Immagine che contiene testo, diagramma, schermata, Diagramma

Descrizione generata automaticamenteImmagine che contiene testo, diagramma, schermata, Diagramma

Descrizione generata automaticamenteFurthermore, the scatter plots highlight the presence of four clusters for each class.



## Laboratory 3 – Dimensionality reduction

1. By applying PCA, we get these histograms of the six different projected features, in descending order of explained variance:Immagine che contiene testo, schermata, diagramma, Diagramma

   Descrizione generata automaticamenteImmagine che contiene testo, diagramma, schermata, Diagramma

   Descrizione generata automaticamenteImmagine che contiene testo, diagramma, schermata, mappa

   Descrizione generata automaticamenteImmagine che contiene testo, schermata, diagramma, Diagramma

   Descrizione generata automaticamenteImmagine che contiene schermata, diagramma, testo, Diagramma

   Descrizione generata automaticamenteImmagine che contiene schermata, diagramma, Diagramma, testo

   Descrizione generata automaticamente

We observe that these dimensions still present histograms with a significant overlapping, except for the first one, where the class distribution are more distinguishable. Moreover, these plots show that, despite the strong overlapping, all the features have a Gaussian distribution, which could lead to an advantage in classification stage.

The scatterplots do not highlight the presence of clusters among reduced features, differently by the ones referred to the original ones.

1. By applying 1-dimensional LDA, we get the following histogram, which refers to the class distribution of the projected feature:

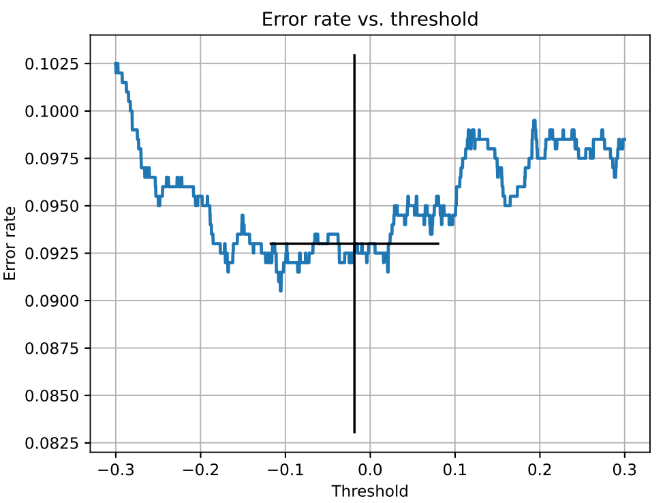
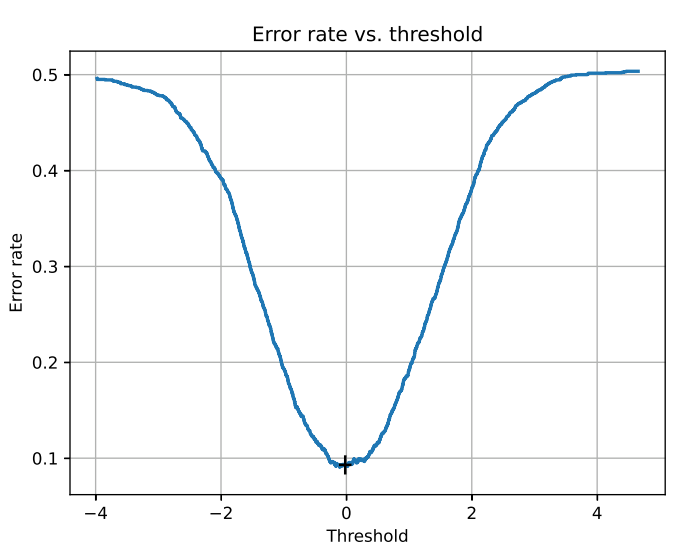
Immagine che contiene testo, schermata, diagramma, Diagramma

Descrizione generata automaticamente

We can observe that a significant overlapping is still present, but:

* This overlapping is less than the one observed in the original features: it is comparable with the one obtained in the first PCA feature.
* both the class distributions are bell-shaped, as after applying PCA.

thus, the LDA direction is better than the original features and it is more suitable for utilization in a classification task.

1. To apply LDA as a classifier, the dataset is split randomly into training and validation set, with a dimension of two-thirds and one-third with respect to the original dataset, respectively. The threshold is computed as the mean of the mean values per class, working only on the training set: its value is about -0.02, as it is possible to notice in the plots at point four. The error rate obtained is 0.093 (9.3 %): the accuracy is greater than 90%, thus the classifier could be considered as sufficiently accurate, if we consider its simplicity.
2. We change the threshold in a wide range, delimited by the minimum and maximum value in the validation set with a step of about 10-4; the goal is to find the value that minimizes the error rate, just by performing the prediction stage with different values of the threshold. The results are shown in the following plots:

The left plot presents a global view: we can observe that the error rate has a minimum around a point (denoted by the black cross), which represents the optimal value of the threshold; it is close to the computed one at the previous point. In fact, by zooming in (right plot), we can notice that the minimum value for the error rate is about 0.09 (9 %) and it corresponds to a threshold value near to -0.1, while the computed one is about -0.02: this is coherent with the error rate obtained as the previous point (0.093), which is near to the optimal one.

Therefore, the accuracy can be improved by a little quantity, but the error rate computed at the previous point is higher than the minimum one by less than 1%.

1. Applying PCA as a preprocessing stage means to:

* Select the number of PCA dimensions *m*, to keep after PCA transformation; in particular, the *m* dimensions chosen are the ones that explain the higher variance.
* Apply 1-dimensional LDA on the dataset, projected over the *m* PCA features.
* Use LDA as a classifier, comparing the results obtained for different values of *m*.

This task has been implemented by iterating over valid (and meaningful) values of *m*, such as from 2 to 5 (included):

* with *m* = 1 LDA would become irrelevant since it would not perform a dimensionality reduction.
* with *m* = 6 PCA would become irrelevant since it would not perform a dimensionality reduction, but only a projection over other directions than the original ones.

The results obtained are the following:

|  |  |  |
| --- | --- | --- |
| PCA dimensions (*m*) | Error rate | Error rate (%) |
| 5 | 0.0930 | 9.30 |
| 4 | 0.0925 | 9.25 |
| 3 | 0.0925 | 9.25 |
| 2 | 0.0925 | 9.25 |

Therefore, we can observe that PCA preprocessing is beneficial for LDA classification in all cases, with an accuracy improvement for each value of *m*, except for five.

## Laboratory 4 – Gaussian density estimation

To perform Gaussian estimation, original dataset is split in two parts (by class), for each feature; then, the best Gaussian fitting is estimated by computing mean and variance with the maximum likelihood method and then by plotting the corresponding Gaussian distribution over the feature histogram. The results obtained are the following:

Immagine che contiene schermata, testo, diagramma, Diagramma

Descrizione generata automaticamenteImmagine che contiene testo, diagramma, schermata, Diagramma

Descrizione generata automaticamenteImmagine che contiene testo, diagramma, schermata, Diagramma

Descrizione generata automaticamente

Immagine che contiene testo, schermata, diagramma, Diagramma

Descrizione generata automaticamenteImmagine che contiene testo, schermata, Diagramma, diagramma

Descrizione generata automaticamente

Immagine che contiene schermata, testo, Diagramma, diagramma

Descrizione generata automaticamente

We can observe that a Gaussian fitting is good only for the first four features, while the other two ones have a different kind of distribution; therefore, this method results to be accurate only to model the first four features. The last two have the same mean value as the estimated Gaussian distribution, but they differ in shape from it.

## Laboratory 5 - Gaussian models

The application of MVG model on the dataset is performed by using the same splits as LDA, in three different variants (standard MVG, Tied MVG and Naïve Bayes MVG); the following error rates are obtained:

|  |  |
| --- | --- |
| Classifier | Error rate (%) |
| Standard MVG | 7.00 |
| Tied MVG | 9.30 |
| Naïve Bayes MVG | 7.20 |

As expected, we observe that:

* Tied MVG classifier has the same error rate as LDA classifier, since they are equivalent models.
* Naïve Bayes classifier is slightly worse than MVG, indeed per-class correlation matrices show a low correlation among dataset features: since covariances are far smaller than feature variances, covariance matrices are almost diagonal, meaning that they have non-diagonal elements, but they are negligible due to their value. Thus, a diagonal representation of per-class covariance matrices is a good approximation of actual ones.

Moreover, we can observe that dataset points seem to be not well-separable with linear surfaces, since LDA performs worse than other two classifiers.

As we can see in the previous plots, the Gaussian assumption holds only for both classes of features 1,2,3,4, whereas features 5,6 have per-class distributions that do not fit well with Gaussian estimation, thus a MVG classification performed only on them will surely lead to inaccurate results.

By applying the same classifiers on features 1-4 (that is discarding the worst-fitted two), we obtain the following results:

|  |  |
| --- | --- |
| Classifier | Error rate (%) |
| Standard MVG | 7.95 |
| Tied MVG | 9.50 |
| Naïve Bayes MVG | 7.65 |

Despite the last two features seem to be misleading for a MVG classifier, they carry some significant information on data distribution, since we observe an increase of error rates by removing them; moreover, since Naïve Bayes performs slightly better than a full covariance MVG, we can conclude that some overfitting has verified: the last two features help removing them, let the MVG performing better than Naïve Bayes, as expected.

By repeating the classification only on features 1-2, we obtain the following results:

|  |  |
| --- | --- |
| Classifier | Error rate (%) |
| Standard MVG | 36.50 |
| Tied MVG | 49.45 |
| Naïve Bayes MVG | 36.30 |

We can observe the following:

* Since features 1-2 classes are not well-separable (as shown in the scatter plot below), all MVG classifiers perform far worse than when applied on all the features.
* Tied MVG classifier performs worse than others since the two features have different per-class variances, thus per-class covariance matrices are surely different, and a tied covariance assumption is inaccurate.

By repeating the classification on features 3-4, we obtain the following results:

|  |  |
| --- | --- |
| Classifier | Error rate (%) |
| Standard MVG | 9.45 |
| Tied MVG | 9.40 |
| Naïve Bayes MVG | 9.45 |

We can observe the following:

* Since features 3-4 classes are well separable (as shown in the scatter plot below), error rates do not increase significantly with respect to the model trained on all features.
* Since per-class variances are similar, tied covariance assumption works well; moreover, the simplification of the model is beneficial, indeed this variant performs slightly better than full-covariance MVG, avoiding the risk of overfitting.

Immagine che contiene schermata, testo, Policromia, diagramma

Descrizione generata automaticamenteImmagine che contiene schermata, testo, diagramma, Policromia

Descrizione generata automaticamenteFinally, applying the same model with a preprocessing stage, implemented as a PCA application on the whole dataset, we obtain the following results, in terms of percentual error rates:

|  |  |  |  |
| --- | --- | --- | --- |
| PCA components | Standard MVG | Tied MVG | Naïve Bayes MVG |
| 2 | 8.80 | 9.25 | 8.85 |
| 3 | 8.80 | 9.25 | 9.00 |
| 4 | 8.05 | 9.25 | 8.85 |
| 5 | 7.10 | 9.30 | 8.75 |

We can observe a general decrease of the accuracy for both Naïve Bayes and standard MVG classifier, due to the reduction of the information extracted from data. Tied MVG performs slightly better for small number of components (less than 5): this is possible since all PCA features (except for the second and, partially, the fifth) have similar per-class covariances, thus the tied covariance assumption is slightly better in this case than in the case without PCA preprocessing.

In conclusion, the best overall Gaussian model (in terms of error rate) is the standard MVG classifier, with an error rate of 7 % on the validation set.

## Laboratory 7 – Model evaluation e Bayes decisions

Model evaluation is performed by computing DCF (both actual and minimum values) considering a set of given application priors, and their corresponding prior log-odds, after having performed label predictions basing on optimal Bayes decisions.

First, the analysis is focused on different application triplets:

* (0.5, 1.0, 1.0): uniform prior and costs
* (0.9, 1.0, 1.0): the prior probability of a genuine sample is higher
* (0.1, 1.0, 1.0): the prior probability of a fake sample is higher
* (0.5, 1.0, 9.0): the prior is uniform, but the cost of accepting a fake image is larger
* (0.5, 9.0, 1.0): the prior is uniform, but the cost of rejecting a legit image is larger

The corresponding effective prior are shown in the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| Prior | False negative cost | False positive cost | Effective prior |
| 0.5 | 1.0 | 1.0 | 0.5 |
| 0.9 | 1.0 | 1.0 | 0.9 |
| 0.1 | 1.0 | 1.0 | 0.1 |
| 0.5 | 1.0 | 9.0 | 0.1 |
| 0.5 | 9.0 | 1.0 | 0.9 |

We can observe that different costs of misclassification are reflected into the effective prior: stronger security value, represented by a high false positive cost, corresponds to a lower prior probability of a genuine fingerprint. Indeed, we can also notice that some of these applications are equivalent, due to the reason just explained, thus we are going to analyze MVG classifier performance on three different applications, given only their effective priors (0.1, 0.5, 0.9), and considering unitary misclassification costs.

Optimal Bayes decisions are computed, in terms on both actual DCF and minimum DCF; the results are shown in the following table:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Effective prior | PCA dimensions | Full covariance | | Tied covariance | | Naïve Bayes | |
| Min. DCF | Act. DCF | Min. DCF | Act. DCF | Min. DCF | Act. DCF |
| 0.1 | 2 | 0.353 | 0.388 | 0.363 | 0.396 | 0.356 | 0.387 |
| 3 | 0.356 | 0.388 | 0.368 | 0.408 | 0.365 | 0.395 |
| 4 | 0.301 | 0.353 | **0.361** | 0.403 | 0.361 | 0.397 |
| 5 | 0.274 | 0.304 | 0.365 | 0.405 | 0.354 | 0.393 |
| Not applied | **0.263** | 0.305 | 0.363 | 0.406 | **0.257** | 0.302 |
| 0.5 | 2 | 0.173 | 0.176 | **0.179** | 0.185 | 0.171 | 0.177 |
| 3 | 0.173 | 0.176 | 0.183 | 0.185 | 0.175 | 0.180 |
| 4 | 0.154 | 0.161 | 0.182 | 0.185 | 0.172 | 0.177 |
| 5 | 0.133 | 0.142 | 0.181 | 0.186 | 0.174 | 0.175 |
| Not applied | **0.130** | 0.140 | 0.181 | 0.186 | **0.131** | 0.144 |
| 0.9 | 2 | 0.438 | 0.443 | 0.435 | 0.479 | 0.432 | 0.442 |
| 3 | 0.439 | 0.468 | **0.434** | 0.457 | 0.434 | 0.459 |
| 4 | 0.415 | 0.460 | 0.444 | 0.462 | 0.431 | 0.463 |
| 5 | 0.351 | 0.398 | 0.445 | 0.463 | 0.431 | 0.466 |
| Not applied | **0.342** | 0.400 | 0.442 | 0.463 | **0.351** | 0.389 |

Since scores are not necessarily well-calibrated, quality comparisons are performed basing on minimum DCF:

* Separately for each application, full covariance model is better than tied covariance model: in particular, this result is clear for π = 0.5 and π = 0.9; moreover, Naïve Bayes model is almost equivalent to full covariance, except for some cases of PCA preprocessing (with a high number of components) where full covariance performs better
* Separately for each variant:
  + Full covariance performs better without PCA preprocessing since some information is lost while reducing dimensionality.
  + Tied covariance performs better with PCA preprocessing (with different number of components, depending on the application): here, PCA helps finding features with similar covariances, so that tied model is more suitable.
  + Naïve Bayes variant performs better without PCA preprocessing, like full covariance model.
* Moreover, relative performance results are consistent for different application: in general, full covariance model performs better than tied covariance model and slightly better than Naïve Bayes one, independently from the application chosen. More details are explained in the previous points.

Scores are not explicitly calibrated: the relative mis-calibration loss is computed as the percentual loss of DCF value, divided by the value of minimum DCF:

Its values are shown in the following table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Effective prior | PCA components | Full covariance | Tied covariance | Naïve Bayes |
| 0.1 | 2 | 10.00 | 9.09 | 8.63 |
| 3 | 8.86 | 10.89 | 8.36 |
| 4 | 17.17 | 11.66 | 9.84 |
| 5 | 11.07 | 11.03 | 10.87 |
| Not applied | 16.06 | 11.91 | 17.59 |
| 0.5 | 2 | 1.68 | 3.45 | 3.48 |
| 3 | 1.42 | 1.08 | 3.04 |
| 4 | 4.69 | 1.60 | 3.09 |
| 5 | 6.59 | 2.69 | 0.75 |
| Not applied | 7.50 | 2.64 | 9.76 |
| 0.9 | 2 | 1.15 | 9.96 | 2.33 |
| 3 | 6.56 | 5.14 | 5.70 |
| 4 | 10.79 | 3.92 | 7.35 |
| 5 | 13.33 | 3.91 | 7.37 |
| Not applied | 16.87 | 4.63 | 10.92 |

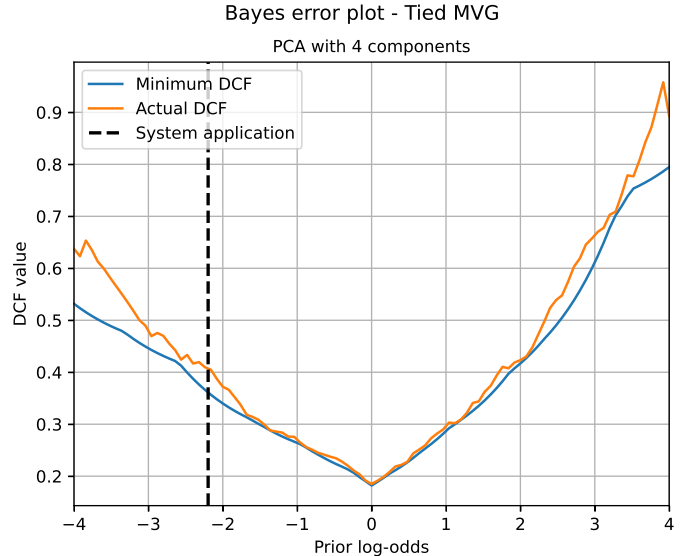
As we can see, tied covariance model is in general better calibrated than others, with full covariance that has a strong calibration loss for applications with unbalanced classes; moreover, we can notice that in general miscalibration increases with the quality of the model, in terms of minimum DCF: it is not a strict rule, but it is an observed trend in all models.

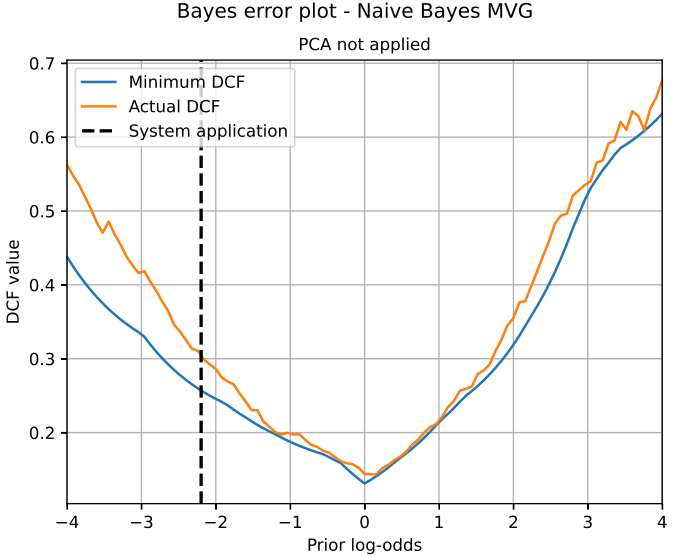
Finally, we can observe that unbalanced class priors are associated with a strong miscalibration, which is more evident with π = 0.1. Scores are not well-calibrated for each application: this is caused by lack of precision in Gaussian estimation related to data distribution, since MVG produces LLR scores, which represent logarithmic ratios between class conditional distributions.

By considering only the first application (π = 0.1), the best results in terms of minimum DCF are the following:

* Full covariance: 0.263
* Tied covariance: 0.361.
* Naïve Bayes: 0.257

Immagine che contiene testo, Diagramma, diagramma, linea

Descrizione generata automaticamenteBayes error plots are the following:

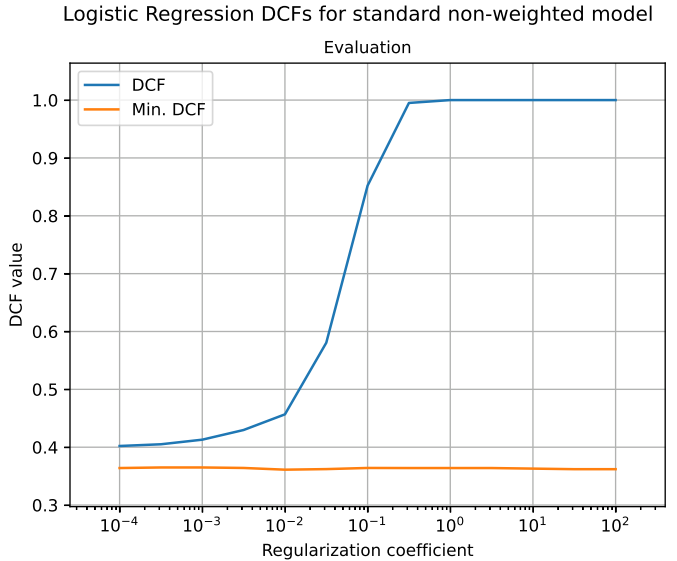


As we can see, Bayes error plots have the same shape in terms of minimum DCF, across the considered range, even if the minimum value is slightly different; the biggest difference is the more significant growth of minimum DCF for tied covariance model, while moving towards applications with unbalanced priors.

Scores are not well-calibrated on the full range of effective prior log-odds: as we can observe, there is a significantly increasing miscalibration moving away from the center of the range; this phenomenon involves our system application (highlighted in the plots), whose effective prior log-odds is:

## Lab 8 – Logistic regression

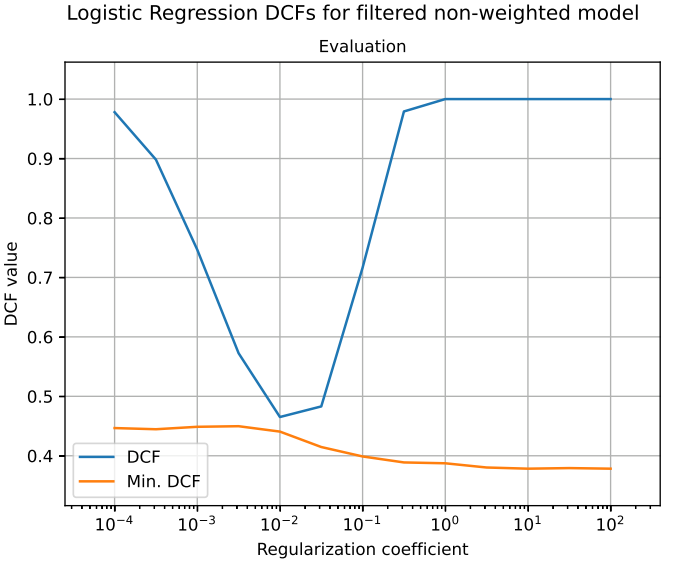
The first task is employed by using an unweighted logistic regression, using different values as regularization coefficient; by considering an application prior πT = 0.1, minimum DCF and actual DCF are measured, and the following results are obtained:



As we can see:

* Minimum DCF is almost invariant to regularization, indeed its value is always included in the interval 0.35 – 0.4: the best one is 0.361, corresponding to λ = 0.01
* Actual DCF is strongly influenced by regularization: its value grows quickly, saturating to one for regularization coefficient values greater than 0.01; it is caused by the loss of probabilistic meaning of this metric when strong regularization is employed. Actual DCF corresponding to best minimum DCF is 0.457. Thus, miscalibration is worse for higher values of regularization, for the reason previously mentioned.
* No significant overfitting phenomena are observed, even with lower values of regularization coefficient, since actual DCF does not increase for smaller values of λ and minimum DCF is almost constant over the range of observed values.

Since overfitting is more relevant for smaller datasets (given the number of parameters to estimate), the same task is performed on a filtered training dataset, with only fifty samples (instead of four thousand); the results are shown in the following plot:



We can observe that:

* Both actual and minimum DCF grow for lower values of λ: in particular, minimum DCF is an almost-decreasing curve, so that it stays constant for smaller values of λ, than it decreases until λ = 10 (0.378), remaining constant again until the end of the range. This is evidence of an overfitting phenomenon, which is mitigated only with strong regularization.
* As in the previous task, actual DCF loses its probabilistic meaning when strong regularization is used.

By using a prior-weighted variant, the information about the application prior is used both to build the model and to compute the scores in prediction step; the following results are obtained:

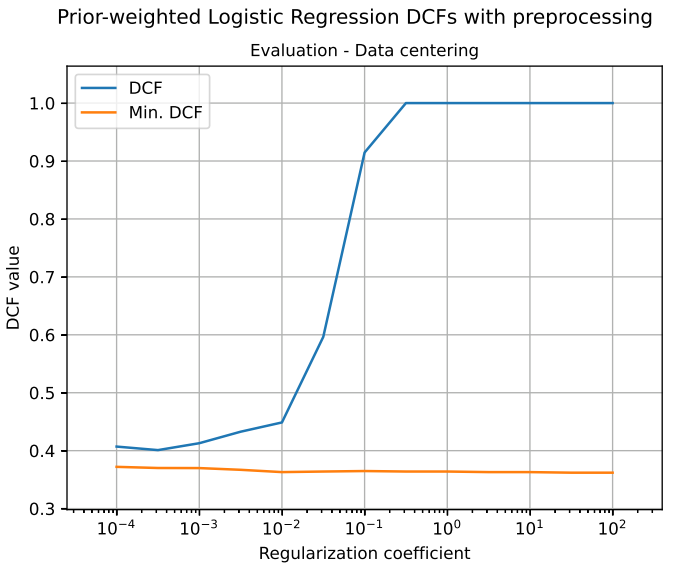
Immagine che contiene testo, linea, Diagramma, diagramma

Descrizione generata automaticamente

We can observe that:

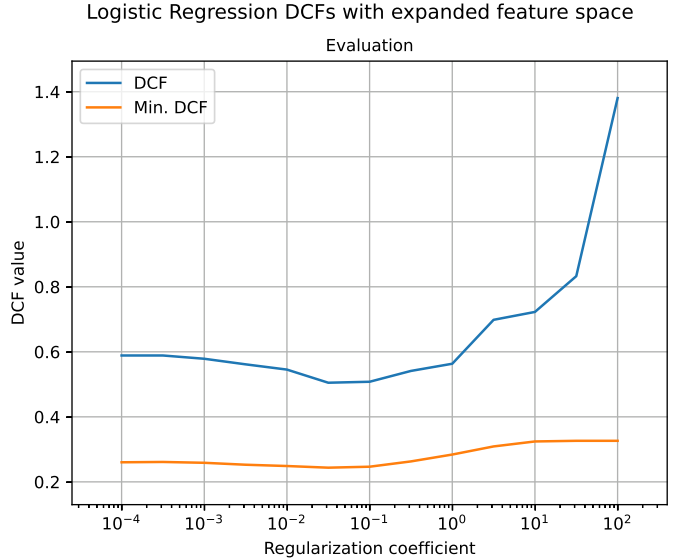
* Actual DCF grows due to loss of probabilistic interpretation, as in the previous tasks.
* Both actual and minimum DCF highlight a slight overfitting phenomenon, since they grow for low values of the regularization coefficient (λ < 0.001)
* The best result is obtained for minimum DCF equal to 0.362, which corresponds to λ = 31.62; since this configuration has a strong regularization, the actual DCF is 1.00, due to loss of probabilistic meaning
* Except for few, not significant differences, the results are like those obtained with an unweighted model.

Since the regularized LR is not invariant to transformations of data features, doing some preprocessing could lead to variations in the results; by centering (both training and validation) data with respect to the training set mean and applying the prior-weighted LR, the following results are obtained:



As we can see, data centering does not significantly affect the results, thus we can observe that data features were already centered with respect to the training set mean; in fact, the obtained plot is the same as the prior-weighted LR without data centering.

By expanding the feature space and applying a prior-weighted LR, it is possible to obtain quadratic boundary surfaces; the results are the following:



We can observe that:

* A phenomenon of overfitting has happened, since actual DCF increases for low values of λ.
* The best result is given by a minimum DCF of 0.244, corresponding to λ = 0.03 and an actual DCF of 0.505
* Data are better separable by a quadratic surface than a linear one since this model works better than a linear LR.

The following table summarizes the best results for MVG and LR classifiers:

|  |  |  |
| --- | --- | --- |
| Model | Minimum DCF | Actual DCF |
| MVG – Full covariance | 0.263 | 0.305 |
| MVG – Tied covariance | 0.351 | 0.403 |
| MVG – Naïve Bayes | 0.257 | 0.302 |
| LR – Unweighted | 0.361 | 0.457 |
| LR – Prior-weighted | 0.362 | 1.000 |
| LR – Prior-weighted expanded | 0.244 | 0.497 |

We can observe that:

* Models that find quadratic boundary surfaces perform better, in terms of minimum DCF.
* MVG models produce scores which are better calibrated than LR models.

thus, we can say that the dataset has features that are better separable by quadratic surfaces than linear ones, and it is coherent with the results obtained in the LR tasks. In conclusion, we observe that a prior-weighted, quadratic logistic regression performs slightly better than a MVG model, even if its results are worse calibrated, having a higher actual DCF than Gaussian models.