Gender Voice Detection

Politecnico di Torino AA 2021 – 2022

Serra Matteo – s303513

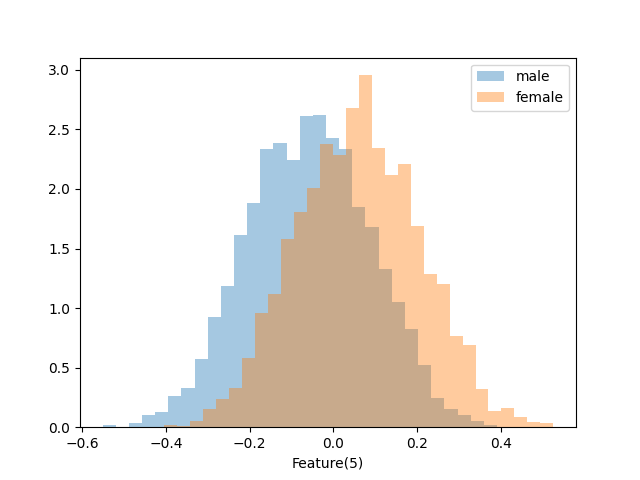
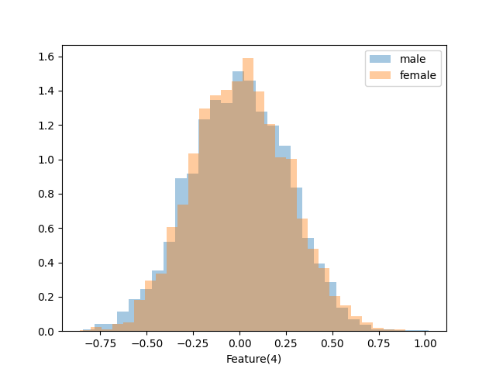
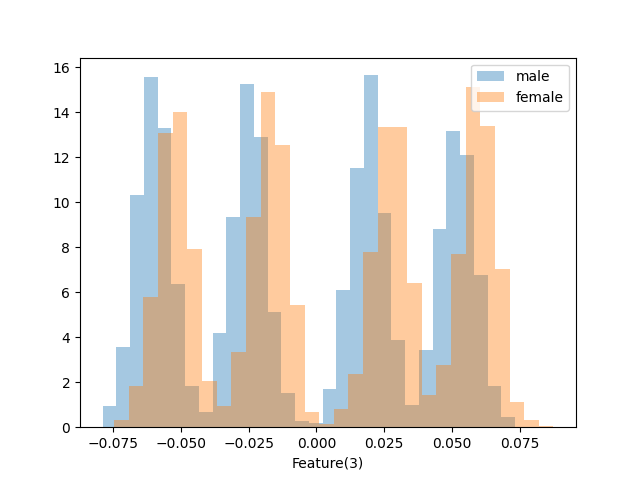
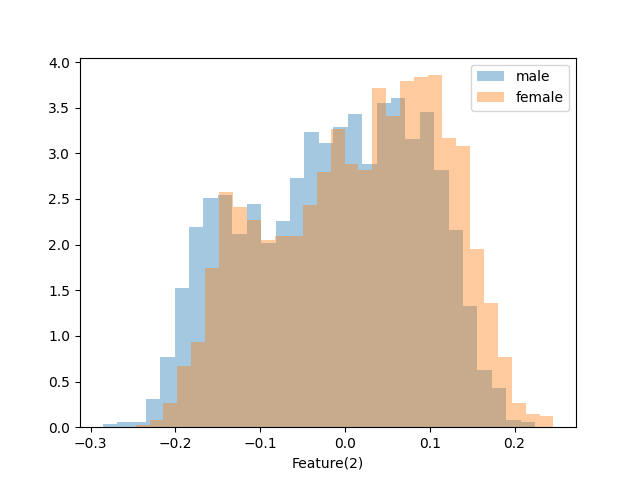
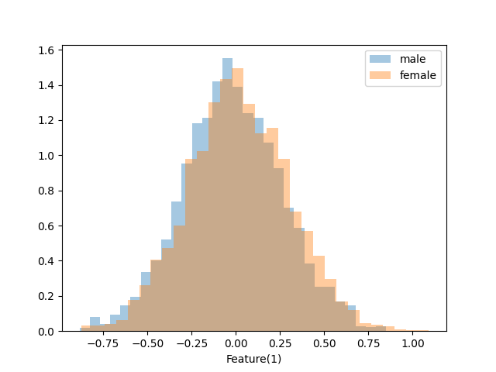
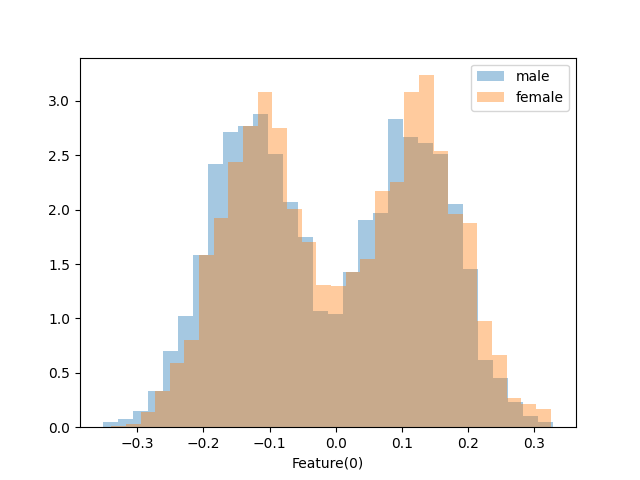


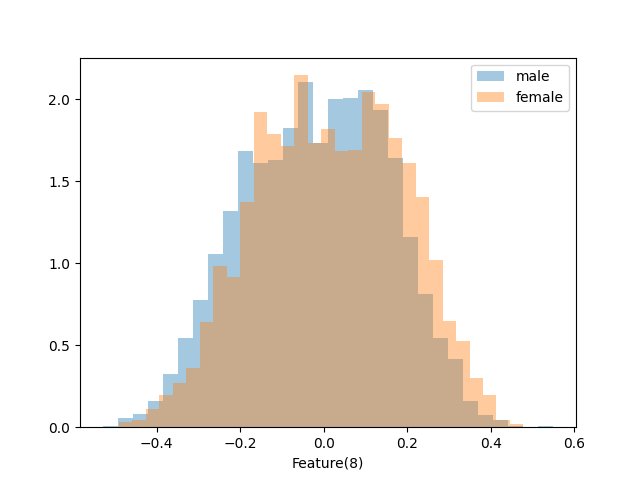
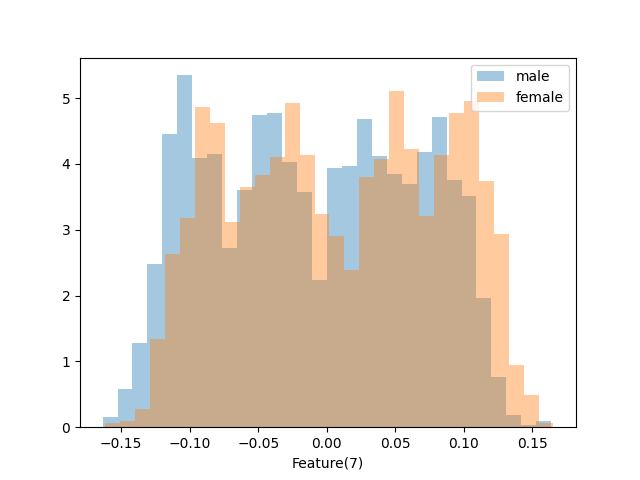
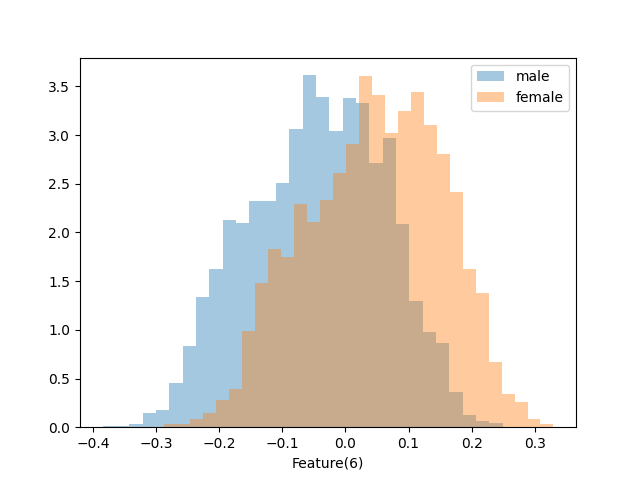
**Introduction**

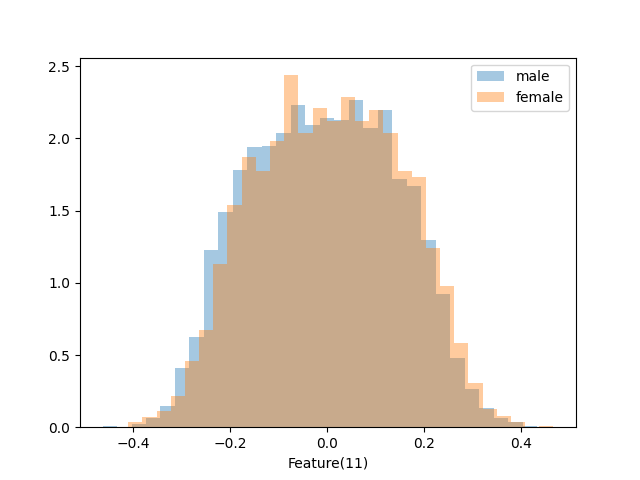
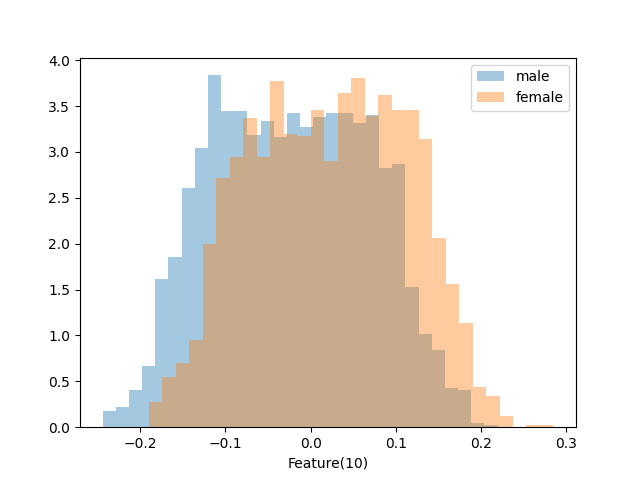
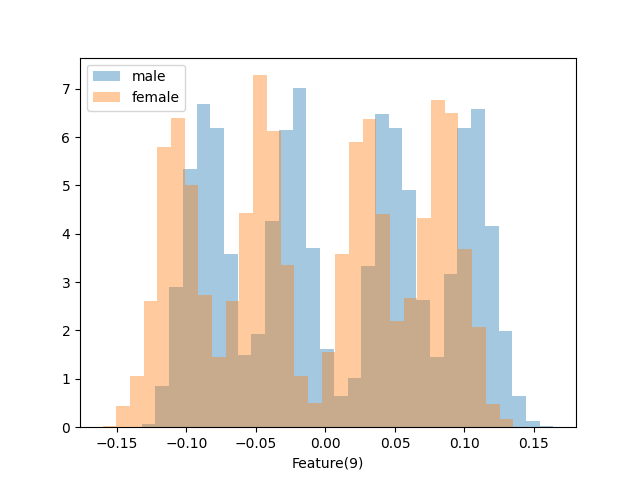
The dataset consists of speaker embeddings that represent the acoustic of a spoken utterance. Each row corresponds to a different speaker and contains 12 features followed by the gender label (1 for female, 0 for male ). The features do not have any particular interpretation. Speakers belong to four different age groups. The age information, however, is not available.

The training set consists of 3000 samples per class, whereas the test set contains 2000 samples per class. To make the problem more tractable, and to avoid potential privacy issues, the dataset consists of synthetic samples that behave similarly to real speaker embeddings. Features are continuous values that represent a point in the *m*-dimensional embeddings space.

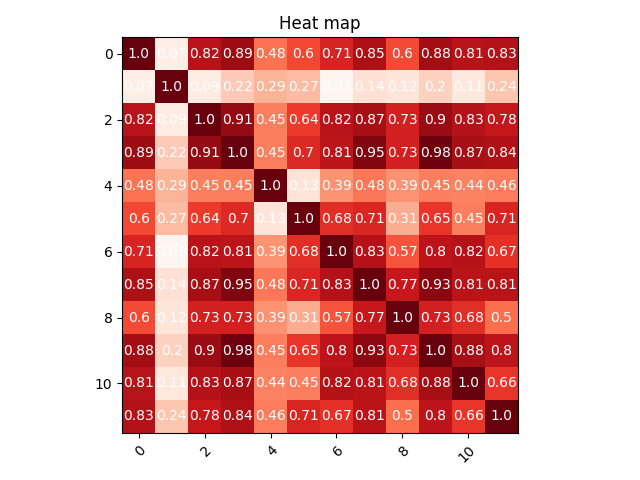
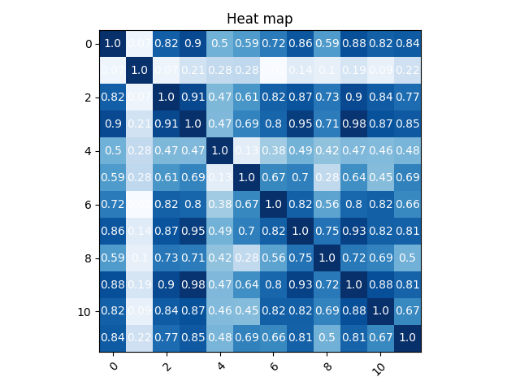
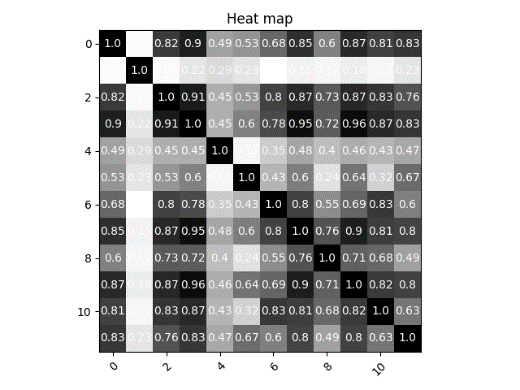
Below we can see the histograms of the dataset features (training set) after the z-normalization.







A correlation analysis of z-normalized features shows that some features are strongly correlated (in this case we assume a strong correlation for values greater than 0.85). Heat maps show the absolute value of the Pearson correlation coefficient: gray: whole dataset, blue: samples of male voices, red: samples of female voices.



This suggests that we may benefit from using PCA to map data to 8 uncorrelated features to reduce the number of parameters to estimate.

Gaussian classifiers

We start considering gaussian classifiers, since within-class covariance matrices are far from being diagonal we will expect not optimal results from the diagonal covariance approach. To understand which model is more promising, and to assess the effectiveness of using PCA we adopt the K-fold cross-validation method with K = 4, in this way the single fold will consist of 75% of the development data and 25% of validation data. Data has been shuffled before splitting, so that the data of different folds are homogeneous.

Our main application will be a uniform prior one:

But we will also consider unbalanced applications

Where the prior is biased towards one of the two classes. In this preliminary analysis we will focus on the min DCF metric, after that we will compute the optimal threshold for each method.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **Z – normalized features – no PCA** | | | |
| Full-Cov | 0.0476 | 0.1246 | 0.1273 |
| Diag-Cov | 0.5593 | 0.8303 | 0.8633 |
| Tied Full-Cov | 0.0467 | 0.1233 | 0.1246 |
| Tied Diag-Cov | 0.5637 | 0.8280 | 0.8553 |
| **Z-normalized features – PCA (m=9)** | | | |
| Full-Cov | 0.3650 | 0.8466 | 0.8300 |
| Diag-Cov | 0.3522 | 0.8267 | 0.8036 |
| Tied Full-Cov | 0.3667 | 0.8557 | 0.8323 |
| Tied Diag-Cov | 0.3550 | 0.8207 | 0.8060 |

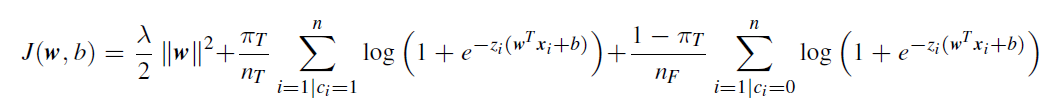
|  |  |  |  |
| --- | --- | --- | --- |
| **Z-normalized features – PCA (m=8)** | | | |
| Full-Cov | 0.4183 | 0.9049 | 0.9090 |
| Diag-Cov | 0.4076 | 0.8793 | 0.8783 |
| **Raw Features – no PCA** | | | |
|  |  |  |  |

Overall, the MVG model with tied covariance performs better, we have to say that the bad results that come from the diagonal covariance models are caused by the strong correlation between features, the diagonal matrices are far from being diagonal. From the table above we can see that the PCA is not effective either for Full or Diagonal covariance models, on the contrary it makes the models results even worse. The full covariance models perform in general slightly worse.

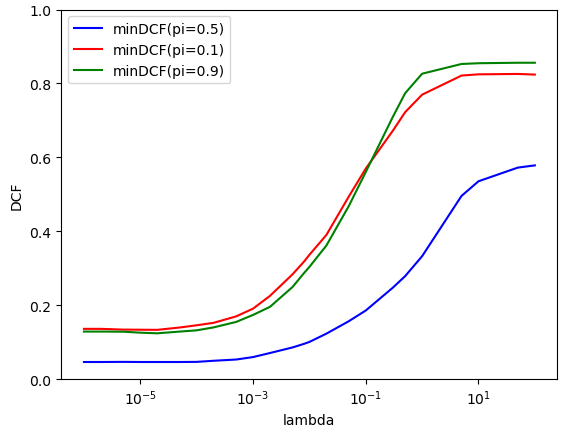
Gaussianization doesn’t improve the classification, so from now we choose to use only the z normalization as preprocessing step.

Overall, the best candidate is currently the MVG model with tied Covariance matrice without PCA and as we can see from the table above the results are also good for imbalanced applications. Given the limited effectiveness of PCA for generative models we only consider using the whole set of features.

Logistic Regression

Our classes are balanced, so re-balancing the cost of the different classes it’s not strictly mandatory. However, we try to re-balanced the costs of the two classes, minimizing: 

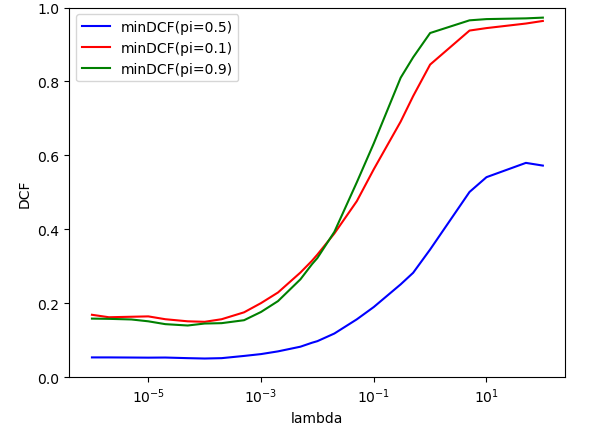
We start considering a prior of and compute the minDCF for each value of lambda in order to tune the hyper-parameter λ. To do that we use a K-fold approach over a validation set, the best value corresponds to the one that have the lowest value of minDCF.



As we can see from the plot the optimal value of lambda is obtained with low value, we select λ= . We can also consider different prior to see the effect on the other applications.

|  |  |  |  |
| --- | --- | --- | --- |
| **Logistic Regression** | | | |
| **4-folds** | | | |
|  |  |  |  |
| Log Reg ( | 0.0463 | 0.1293 | 0.1256 |
| Log Reg ( | 0.0473 | 0.1360 | 0.1277 |
| Log Reg ( | 0.0466 | 0.1300 | 0.187 |
| **MVG** | | | |
| MVG (Full-Cov) | 0.0476 | 0.1246 | 0.1273 |
| MVG (Tied Full-Cov) | 0.0467 | 0.1233 | 0.1246 |

Overall, the MVG model with full covariances perform slightly. Since MVG corresponds to quadratic separation rules, we repeat the analysis for Quadratic Logistic Regression.



Again, we consider training using different prior to see the effects on the other applications.

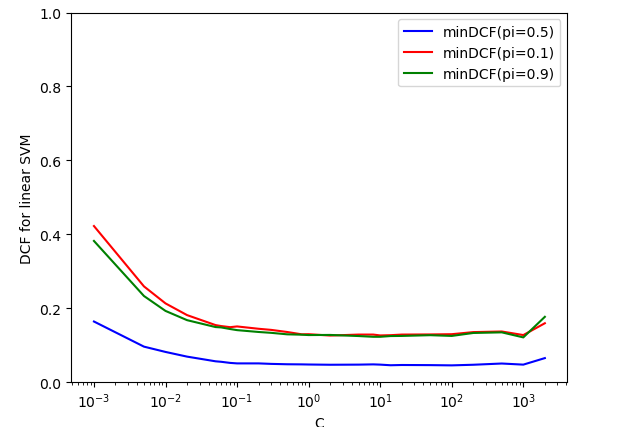
|  |  |  |  |
| --- | --- | --- | --- |
| **Logistic Regression** | | | |
| **4-folds** | | | |
|  |  |  |  |
| Quad Log Reg ( | 0.0533 | 0.1463 | 0.1437 |
| Quad Log Reg ( | 0.0557 | 0.1689 | 0.1500 |
| Quad Log Reg ( | 0.0557 | 0.1490 | 0.1583 |
| **MVG** | | | |
| MVG (Full-Cov) | 0.0476 | 0.1246 | 0.1273 |
| MVG (Tied Full-Cov) | 0.0467 | 0.1233 | 0.1246 |

As before, a good value for lambda is but as we can see from the table above, neither the quadratic logistic regression improves the performances of the log reg model. The min DCF becomes better for those applications with unbalanced class distribution, but not good enough to outperform the MVGs classifiers.

SVM

Linear SVM

For linear SVM, we need to tune the hyper-parameter C. Again, we use the k-fold cross validation method to find the near-optimal value of C. We start with a model that does not balance the two classes.



To rebalance the classes, we use a different value of C for the different classes

Immagine che contiene testo, orologio

Descrizione generata automaticamente

Subject to

Immagine che contiene orologio

Descrizione generata automaticamente

Where the i-th C corresponds to for samples of the class , or to for samples of the other class. Since we are not modelling the bias term, we omitted the constant related to the bias. So, we select and where and are the empirical priors for the two classes computed over the training set.

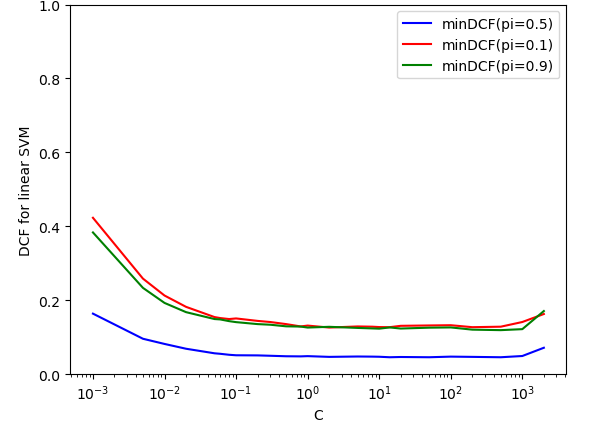
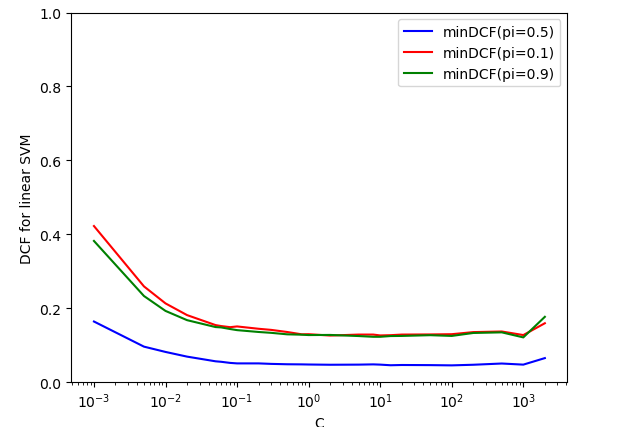


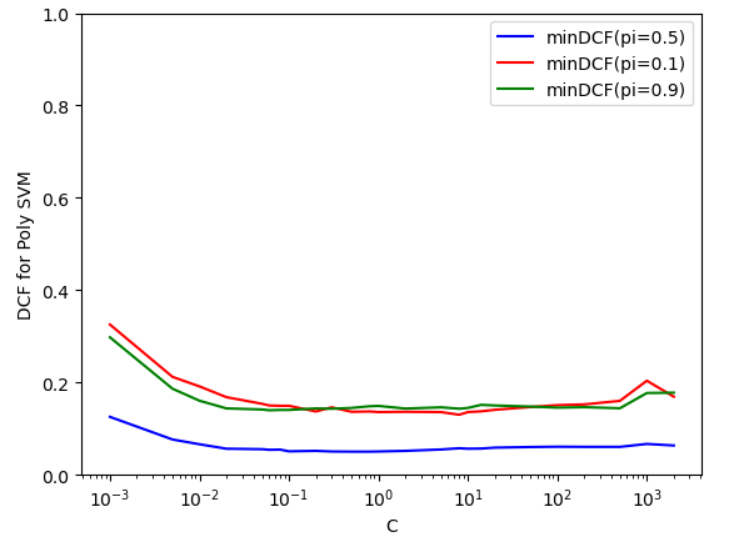
Figure 1-linear SVM without class rebalancing Figure 2 - linear SVM with class rebalancing

As we can see from the graphics above the class rebalancing doesn’t improve the performance of the linear model. The choice of C does not look critical, we select C = 1 because from that point the value of the duality gap starts to increase, and the approximation becomes less accurate. We can compare linear models in terms of min DCF:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| MVG (Tied Full-Cov) | 0.0467 | 0.1233 | 0.1246 |
| Log Reg ( | 0.0463 | 0.1293 | 0.1256 |
| Linear SVM (C= 1) | 0.0476 | 0.1297 | 0.1273 |
| Linear SVM (C= 1, ) | 0.0476 | 0.1297 | 0.1273 |

RBF and Poly SVM

We are going to consider two non-linear SVM models. The first will use a polynomial quadratic kernel, for this one we expect similar results respect the Quadratic Logistic Regression models. The second will employ a radial basis function kernel. For the quadratic kernel we have to estimate the value of C, in order to do this, we use the k-fold approach and evaluate the min DCF for each value of C.



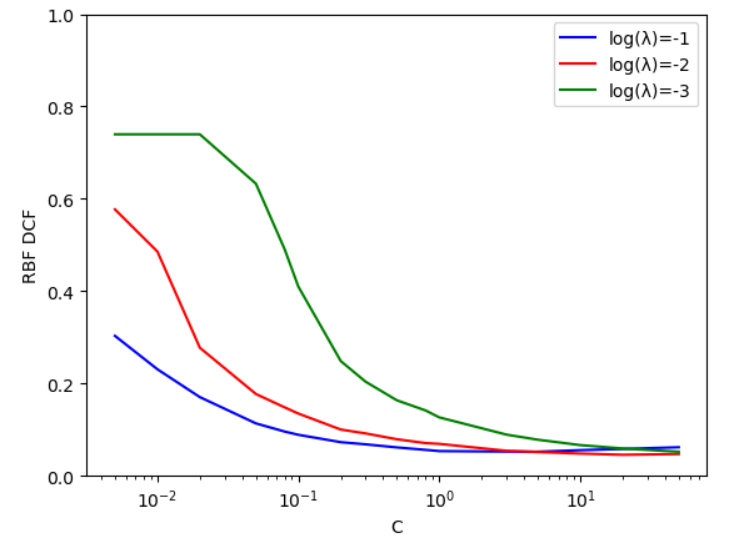
Choice of C does not look critical for this reason we chose C = 0.1.

We can now compare quadratic models in terms of minDCF

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| MVG (Full Cov) | 0.0476 | 0.1246 | 0.1273 |
| Quad Log Reg ( | 0.0533 | 0.1463 | 0.1437 |
| Quadratic SVM (C = 0.1) | 0.0507 | 0.1487 | 0.1407 |

RBF SVM

In this analysis we try different values of C and different combinations of γ.



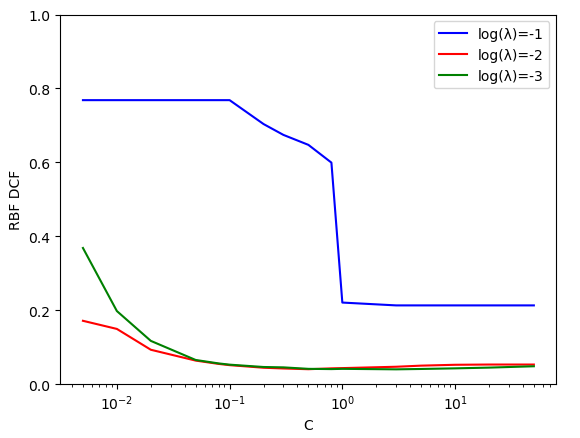
The plot shows that both γ and C influence the results. Best results are obtained using log γ = -2 and C=10. Class re-balancing for the primary task (using the same values for C and γ as for imbalanced model) provides very similar result since the dataset is already class balanced.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| MVG (Full-Cov) | 0.0476 | 0.1246 | 0.1273 |
| Quad Log Reg ( | 0.0533 | 0.1463 | 0.1437 |
| Quadratic SVM (C = 0.1) | 0.0507 | 0.1487 | 0.1407 |
| RBF SVM (C = 10, log γ = -2) | 0.0480 | 0.1510 | 0.3900 |

We try with class rebalancing in order to see if it helps the performance on different applications.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| RBF SVM (C = 10, log γ = -2) | 0.0480 | 0.1510 | 0.3900 |
| RBF SVM (C = 10, log γ = -2, ) | 0.0480 | 0.1480 | 0.1253 |
| RBF SVM (C = 10, log γ = -2, ) | 0.0503 | 0.1600 | 0.1373 |
| RBF SVM (C = 10, log γ = -2, ) | 0.0527 | 0.1570 | 0.1417 |

We also repeated the analysis using raw features.



As we can see, the chosen gamma is still the better and the choice of C does not look critical Overhales the results are again consistent.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **Z-norm features** | | | |
| RBF SVM (C = 10, log γ = -2) | 0.0480 | 0.1510 | 0.3900 |
| RBF SVM (C = 10, log γ = -2, ) | 0.0480 | 0.1480 | 0.1253 |
| RBF SVM (C = 10, log γ = -2, ) | 0.0503 | 0.1600 | 0.1373 |
| RBF SVM (C = 10, log γ = -2, ) | 0.0527 | 0.1570 | 0.1417 |
| **Raw features** | | | |
| RBF SVM (C = 10, log γ = -2) |  |  |  |
| RBF SVM (C = 10, log γ = -2, ) | 0.0513 |  |  |
| RBF SVM (C = 10, log γ = -2, ) |  |  |  |
| RBF SVM (C = 10, log γ = -2, ) |  |  |  |

GMM

The last model we consider is a generative approach based on training a GMM over the data of each class. GMMs can approximate any sufficient regular distribution to a desired degree. We will try this method on our dataset, always with k-fold, with different value for the components and different covariance approaches. Since the initialization of the initial GMM plays an important role in GMM training we chose to use

the LBG algorithm, so we start from a MVG solution and split to a 2G GMM at each iteration. In

order to find the right number of components we resort to cross validation.