Machine Learning and Pattern Recognition Gender Identification Project

Nicolò Caradonna - s316993

September 2023

1 Introduction

1.1 Abstract

The objective of this project is to create a classifier that can determine the gender of individuals based on high-level characteristics extracted from facial images.

Our dataset contains image embeddings, which are compact representations of images achieved by mapping facial images to a shared, low-dimensional space, typically with just a few hundred dimensions. To maintain the feasibility of our model, we are working with synthetic data, resulting in embeddings with considerably lower dimensions compared to real-world scenarios.

1.2 Consideration on Dataset

The dataset contains **12-dimensional vectors**, consisting of continuous values, belonging to either the **male** category (labeled as 0) or **female** category (labeled as 1). It's important to note that the individual components of these 12-dimensional vectors do not carry any specific real-world meaning so they don't have any physical interpretation.

In the file named **"Train.txt"**, you'll find the embeddings that can be used to construct the classification model, which serves as our training set. On the other hand, in the file labeled **"Test.txt"** you will discover the embeddings used for evaluation purposes, constituting our evaluation set.

Each row in both files represents a single sample.The features of each sample and their corresponding labels are separated by commas within each row. Specifically, the first 12 columns of each file contain the sample components, while the last column holds the corresponding label.

- The training set contains **2400** samples divided in **720** male and **1680** female. We can notice that the training set is very unbalanced with 70% of samples belonging to female class. The samples belong to 3 different age groups. Each age group may be characterized by different distributions for the embeddings, unfortunately the age information is not available.
- The test set contains **6000** samples divided in **4200** male and **1800** female. We can notice that the test set is very unbalanced with 70% of samples belonging to male class, therefore the proportion of samples in training and test sets is very different.

1.3 Feature Analysis

In the following section, some preliminary feature analysis are presented so that we can understand which models might be most suited for our problem.

1.3.1 Histograms and Scatter Plots

Plot of distributions and scatter plots are shown here. Note that before plotting the distributions, data have been centered.

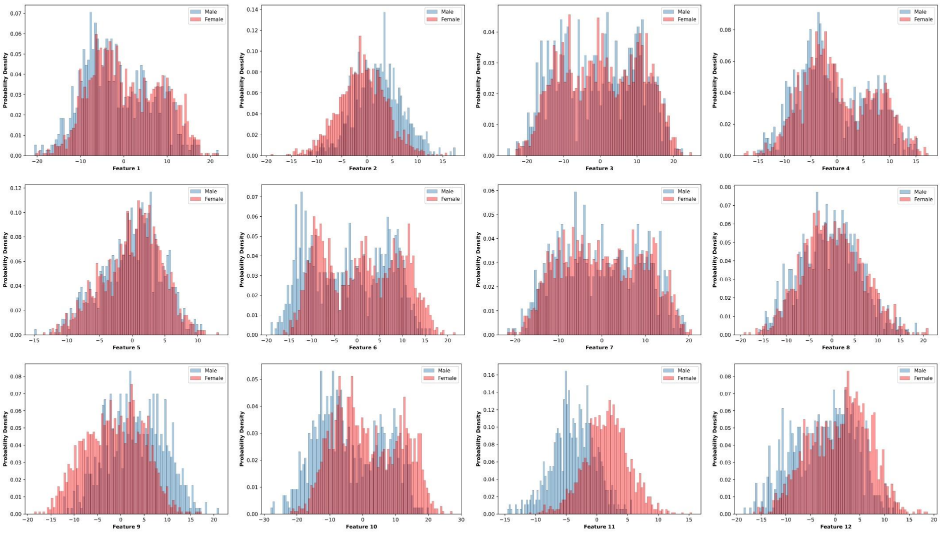


Figure 1: Feature Distributions

From an initial analysis, it is evident that some of these features can be accurately described by Gaussian distributions, particularly features 2-5-8-9-11 have a distribution resembling a one-dimensional Gaussian. Other features, however, appear to be distributed as a composition of multiple Gaussians, this is evident with features 3-6-10 where three Gaussian distributions appear. Probably the reason for this is that there are samples from three different age groups in the dataset and thus we find three different clusters in the distributions. In contrast, there does not seem to be any difference between the distributions of the two classes (blue-male/red-female).

Also from scatter plot analysis, it is possible to see the presence of features composed of only one cluster and features where there are three different clusters. Again the distribution of the

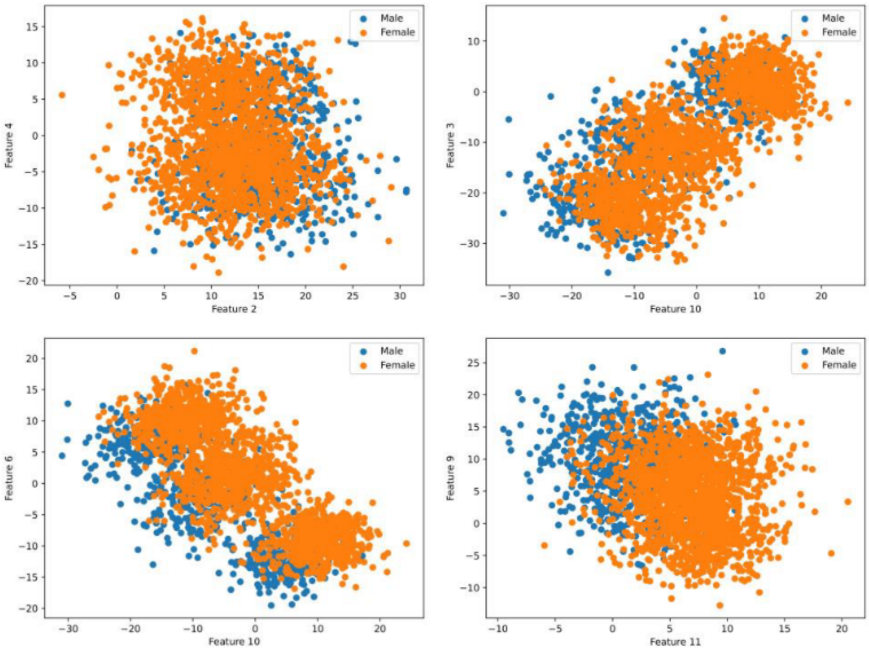


Figure 2: Scatter Plots

samples appears to be a Gaussian so we can assume that a Gaussian model will perform well to classify our data.

1.3.2 Linear Discriminant Analysis

Now we can focus our attention on a preprocessing transformation known as Linear Discriminant Analysis (LDA). The aim of LDA is to find the most discriminant direction that separates the classes of our dataset. Since LDA can find C-1 discriminant directions where C is the number of classes in the dataset, we will have only one discriminant direction. Looking at the graph below, it can be seen that by applying this transformation technique, the data become more linearly separable but nevertheless there is a region where classification errors would still be made.

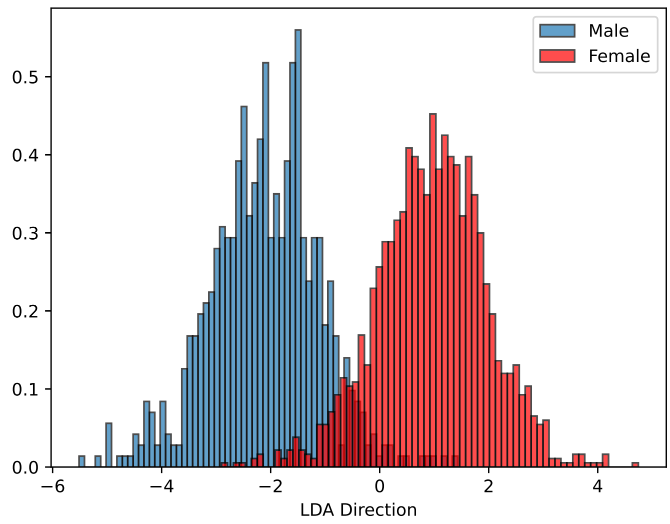
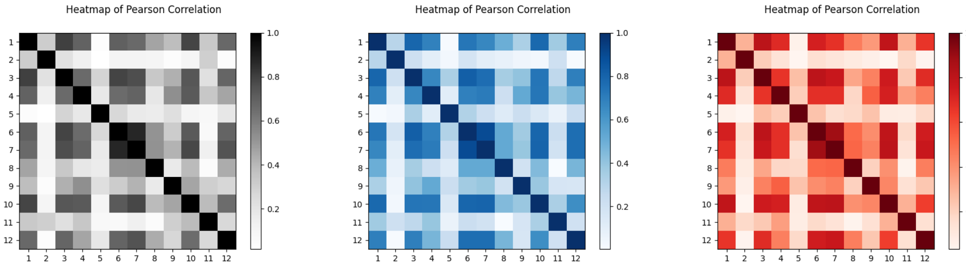


Figure 3: LDA Plot

1.3.3 Pearson Correlation

An additional analysis we can perform on our dataset concerns the correlation between features. To do this we exploit Pearson correlation, showing the results on a heatmap. To better understand the results of the heatmap we should note that the value of the correlation can range from 0 to 1. The higher the value, the higher the correlation between the corresponding features. This information will allow us to develop new hypotheses about some classifiers, which are influenced by feature correlation.



(a) All classes correlation

(b) Male class correlation

(c) Female class correlation

As we can see from the heatmaps, the dataset has slightly correlated features, in particular this correlation is more evident between some features such as between 1-3 , 1-10 and 7-6. In other cases the correlation is totally absent such as between features 1-5 or 2-5. Analyzing individually the correlations within the two classes we can see that the results are the same as in the dataset , in fact the remarkable information is that the heatmaps of the two classes are almost equal. This fact is very important since having the classes correlated in the same way indicates that some classifiers such as MVG and Tied Gaussian will perform similarly. This data is also useful for making assumptions about models such as Naive Bayes where it is assumed that the features are uncorrelated. It is clear from the heatmaps that a Naive Bayes model may not perform so well since a correlation is present, however, it is necessary to further investigate this assumption by directly applying the algorithm.

1.3.4 PCA and Explained Variance

We now focus instead on PCA and in particular on Explained Variance, which allows us to understand for each component identified by PCA, how much variability in the dataset is preserved.

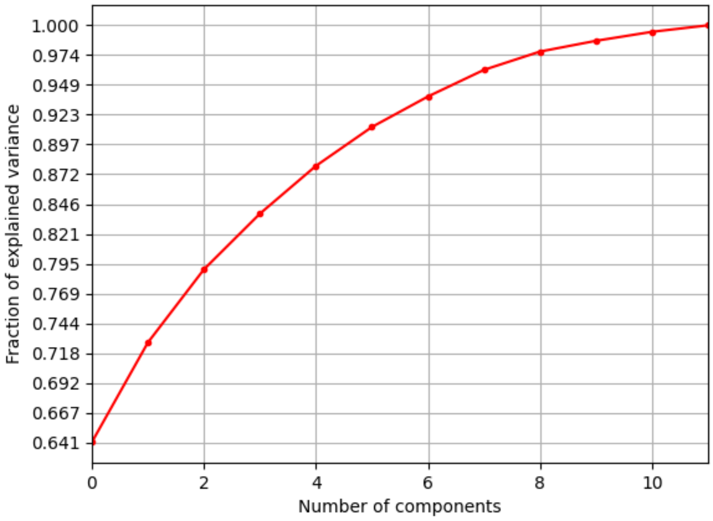


Figure 5: Explained Variance

From the graph we can see that by reducing the size of our dataset from 12 to 8 by PCA, the variance of the dataset still remains above 95% while going from 12 dimensions to 9 the variance of our dataset is above 97%. Decreasing to 9 we will get above 90%.

2 Classification and Training Protocol

2.1 Introduction

The following model have been considered for our classification problem:

1. Generative Models
   - MVG
   - MVG + Diagonal Covariance
   - MVG + Tied Covariance
   - MVG + Diagonal Covariance with Tied Covariance
2. Logistic Regression
   - Prior Weighted Logistic Regression
   - Quadratic Logistic Regression
3. Support Vector Machine
   - Linear SVM
   - Quadratic SVM (polynomial kernel function with degree=2)
   - SVM with Radial Basis kernel Function
4. Gaussian Mixture Models
   - Gaussian Mixture Models (GMM)
   - GMM + Diagonal Covariance
   - GMM + Tied Covariance
   - GMM + Diagonal Covariance with Tied Covariance

For what concerns the **Training Protocol** note that:

- A K-Fold approach has been used with K = 5, for all the following results.
- Preprocessing techniques such as PCA or Z-Norm have been also applyed.
- To assess the best model we refer for the moment, only to **minDCF** metric.
- The target application considers a balanced use-case, with a working point defined by the triplet ($\pi_T = 0.5$,$C_fn = 1$,$C_fp = 1$). Performance of the models for alternative applications are also presented.

2.2 Multivariate Gaussian Classifier

In this section, we are analyzing the Gaussian Classifier and all its versions as said before. Gaussian Classifiers works under the assumption that data follow a Gaussian Distribution:

$$X|C \sim N(\mu_c, \Sigma_c)$$

Our expectations about this classifier are that it can perform well, since we have seen that some features are distribuited as gaussians.

2.2.1 MVG

As we expected, the MVG classifier performs well. Different values of PCA have been tried. With 11-dimensions the result are linear with raw data while decreasing the number of dimensions, results get slightly worse. Among the three application, the target application with no PCA has the best results.

PCA	min_{DCF} $(\pi = 0.5)$	min_{DCF} $(\pi = 0.1)$	min_{DCF} $(\pi = 0.9)$
raw	**0.114**	0.280	0.362
m = 11	0.116	0.288	0.356
m = 10	0.166	0.412	0.473
m = 9	0.183	0.424	0.556
m = 8	0.189	0.449	0.564
m = 7	0.265	0.637	0.704

2.2.2 Naive Bayes

The Naive Bayes assumption, assumes that features are uncorrelated and results depends on how accurate this assumption is. Looking at heatmaps, we have seen that some features are highly correlated with each other, and others that are uncorrelated. The results are in line with that in fact we get a value not very high but definitely worse than the standard MVG. We can also see that applying PCA and then Naive Bayes assumption, better results are given.

PCA	min_{DCF} $(\pi = 0.5)$	min_{DCF} $(\pi = 0.1)$	min_{DCF} $(\pi = 0.9)$
raw	0.459	0.788	0.779
m = 11	**0.132**	0.299	0.371
m = 10	0.172	0.425	0.477
m = 9	0.192	0.444	0.551
m = 8	0.199	0.448	0.553
m = 7	0.277	0.641	0.680

2.2.3 Tied MVG

Tied Gaussian Classifier, assumes a single covariance matrix for all the classes of the dataset. This assumption is incorrect if the classes present different correlations within them. In our case outcomes are in line with MVG ones. This result confirms what we expected, indeed looking at correlation heatmaps, the within class correlation are almost the same.

PCA	min_{DCF} $(\pi = 0.5)$	min_{DCF} $(\pi = 0.1)$	min_{DCF} $(\pi = 0.9)$
raw	**0.116**	0.3299	0.340
m = 11	0.121	0.300	0.355
m = 10	0.165	0.395	0.469
m = 9	0.183	0.413	0.546
m = 8	0.186	0.435	0.554
m = 7	0.268	0.621	0.700

2.2.4 Tied Naive Bayes

Tied Naive Bayes combines the assumptions of the previous two models so the results are in line with those of the Naive Bayes model

PCA	min_{DCF} $(\pi = 0.5)$	min_{DCF} $(\pi = 0.1)$	min_{DCF} $(\pi = 0.9)$
raw	0.462	0.782	0.786
m = 11	**0.122**	0.299	0.361
m = 10	0.174	0.412	0.485
m = 9	0.187	0.432	0.543
m = 8	0.193	0.451	0.551
m = 7	0.267	0.645	0.685

2.3 Logistic Regression

The next model we are going to analyze is a Discriminative model and specifically Logistic Regression. Analysis of the dataset showed that the classes are not balanced so a weighted version of the logistic regression has been implemented:

$$R(w) = \frac{\lambda}{2}\|w\|^2 + \frac{\pi_T}{n_T}\sum_{i|z_i=1}\ell(z_i s_i) + \left(\frac{1-\pi_T}{n_F}\right)\sum_{i|z_i=-1}\ell(z_i s_i) \tag{1}$$

This model provides a linear separation rule, however, it is possible to extend the model to obtain quadratic separation surfaces. The value λ is an hyperparameter of our model and it can be tuned to optimize the performance of the classifier:

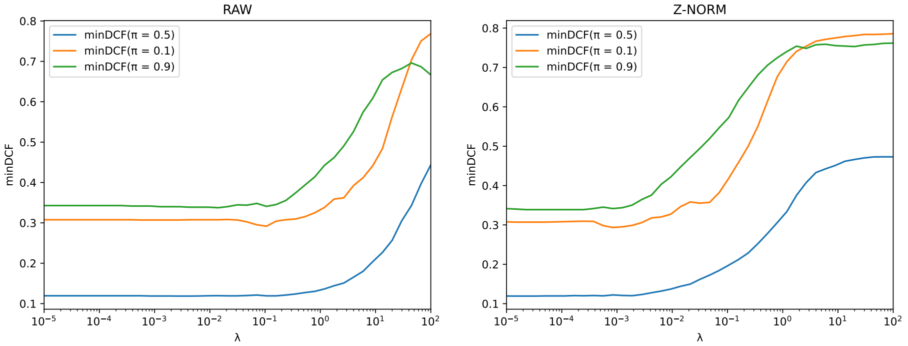
- λ is too large, the model will not well separate the classes.
- λ is too small, we will get a solution that has good separation on the training set, but may have poor classification accuracy for unseen data.

The Tied Gaussian Classifier that we analyzed earlier, introduces a linear separation rule, and we also saw that the results are quite good and in line with the MVG that introduces quadratic classification rule. So the classes in our dataset can be separated by a linear separation rule so we expect the LR results to be similar to those obtained with the TMVG. As before we are going to analyze different applications by applying some preprocessing techinices to see if they are effective. In this section we are also going to analyze the Quadratic Gaussian Classifier, which precisely allows us to exploit the logic of logistic regression but applying separation rules of a quadratic type. In particular, this is possible by mapping the dataset into an expanded feature space. The reason for this choice comes from the fact that the MVG model, which applies quadratic separation rules, has shown good results so we expect the same from this model as well.

2.3.1 Logistic Regression Results

The first analysis we perform by applying LR is to compare the behavior of the model as the lambda hyperparameter changes. In particular, λ values from 10^-5 up to 10^2 were analyzed by comparing different applications as well. The value of π_T chosen initially is 0.5

As we can see, the results are in line with those obtained with TMVG exactly as we expected, moreover we can also observe that as the value of lambda changes, different results are obtained. Going from a lambda value of approximately 10^{-1} to 10^2 results become worse, indicating that our model has become too general and can no longer split the data well. While decreasing the value of λ to zero the performances of the model increses. The lambda value to be chosen is therefore certainly less than 10^{-1} but we cannot go too close to 0 otherwise our model may be subject to overfitting.



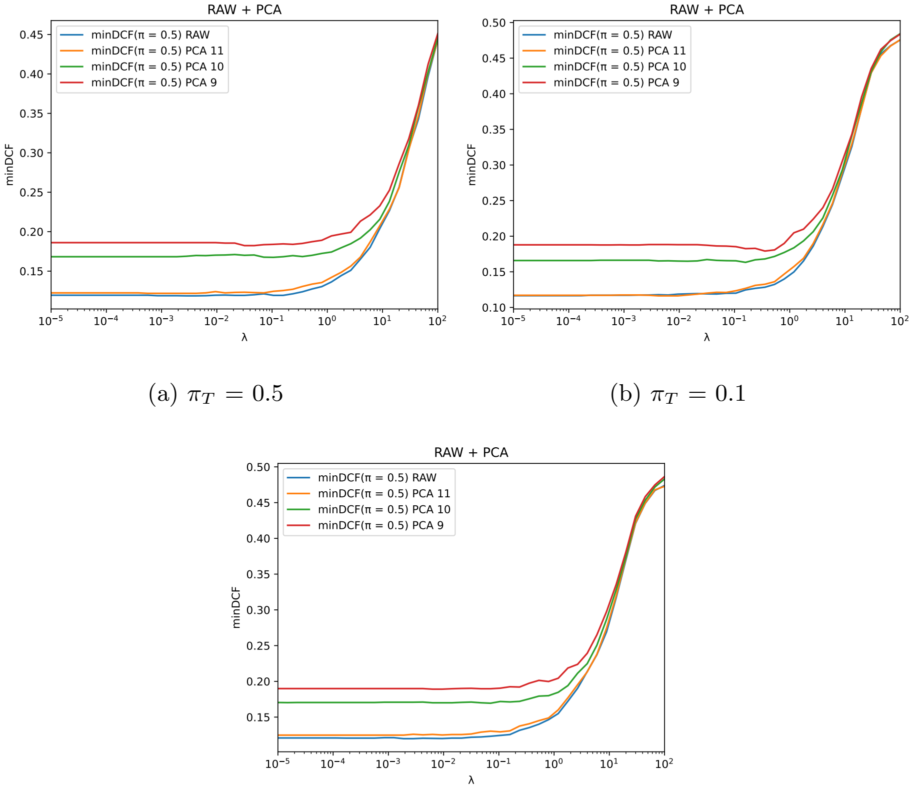
(a) RAW features with $\pi_T = 0.5$

(b) Z-norm features with $\pi_T = 0.5$

In the following table, some values of min_{DCF} with $\lambda = 0$ are showed also for different values of prior π_T. Results obtained with Z-norm have been omitted, since they are very similar. The best result is obtained wit $\pi_T = 0.9$

Model $\lambda = 0$	min_{DCF} $(\pi = 0.5)$	min_{DCF} $(\pi = 0.1)$	min_{DCF} $(\pi = 0.9)$
$\pi_T = 0.5$	0.122	0.295	0.344
$\pi_T = 0.9$	**0.117**	0.321	0.336
$\pi_T = 0.1$	0.121	0.302	0.384

For the sake of completeness, the following graphs are given, in which the model is trained after preprocessing the dataset with various PCA values. From the results obtained, as found in the Gaussian Model, applying PCA does not guarantee any improvement



(a) $\pi_T = 0.5$

(b) $\pi_T = 0.1$

(c) $\pi_T = 0.9$

2.3.2 Quadratic Logistic Regression results

Regarding Quadratic Logistic Regression, the algolithm was tested by considering the target application with a prior $\pi_T = 0.5$, comparing the results by applying Z-norm. From the evidence obtained, it appears that the most promising results are obtained considering a value of λ approximately equal to 10^2 without applying z-norm where instead the best values are obtained for values of λ close to zero.

RAW / $\lambda = 10^2$	min_{DCF} $(\pi = 0.5)$	min_{DCF} $(\pi = 0.1)$	min_{DCF} $(\pi = 0.9)$
$\pi_T = 0.5$	0.126	0.298	0.374
$\pi_T = 0.1$	0.134	0.348	0.348
$\pi_T = 0.9$	0.127	0.301	0.408

Z-norm / $\lambda = 0$	min_{DCF} $(\pi = 0.5)$	min_{DCF} $(\pi = 0.1)$	min_{DCF} $(\pi = 0.9)$
$\pi_T = 0.5$	0.139	0.289	0.348
$\pi_T = 0.1$	0.137	0.328	0.346
$\pi_T = 0.9$	0.144	0.345	0.365

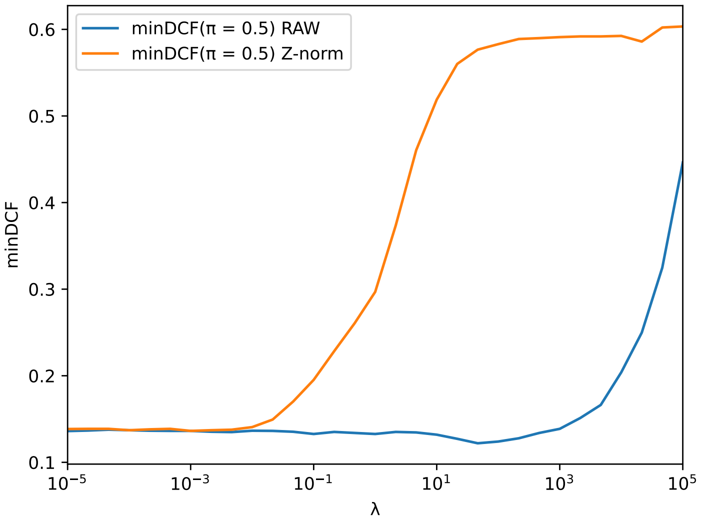


Figure 8: RAW and Z-norm features with $\pi_T = 0.5$

2.4 Support Vector Machine

We are focusing now on a new classifier also called Support Vector Machine, in both its linear and non-linear versions. Just as logistic regression, we are going to consider a weighted version of the problem.

- **Linear SVM**

  We can compute the primal solution by minimizing:

$$J_b(\mathbf{w}_b) = \frac{1}{2}\|\mathbf{w}\|^2 + C_i \sum_{i=1}^{n} \max\left(0, 1 - z_i(\mathbf{w}_b^T \mathbf{x}_i)\right) \tag{2}$$

  where:

  - $C_i = \begin{cases} \frac{C}{\pi_T}\frac{\pi^T_{emp}}{\pi_T} & if \;\; i \in Class0 \\ \frac{C}{\pi_F}\frac{\pi^F_{emp}}{\pi_F} & if \;\; i \in Class1 \end{cases}$

  Note that π_{emp} is the empirical prior of the dataset, obtained as the fraction of samples of a given class among all the samples.

- **Non-Linear SVM**

  The same idea used for Quadratic Logistic Regression can also be used for SVM, specifically by expanding the feature space and then applying the model, nonlinear decision functions can be obtained. This is possible through the use of kernel functions, which allow the dot product between matrices to be computed in the expanded space, without actually transforming the dataset. Kernel functions can be used due to the fact that formulation of the SVM can also be done with a so-called dual solution, we do not go into this topic in depth since this is not the intent of our discussion.

We will analyze:

- **Polynomial kernels of degree d**
- **Gaussian Radial Basis Function kernel**

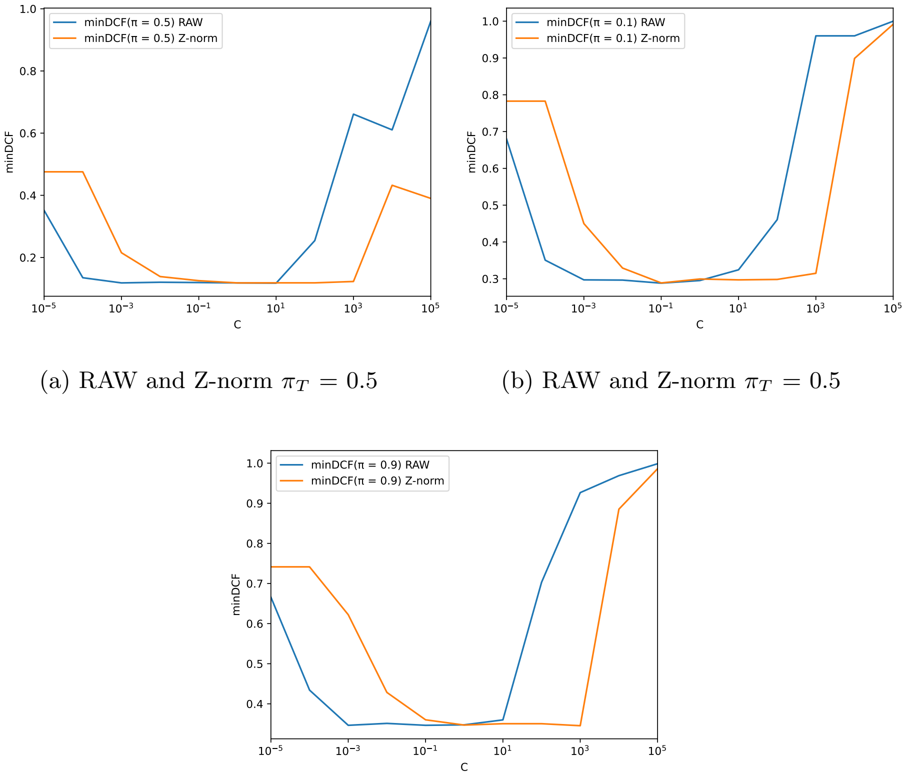
2.4.1 Linear SVM results

As always, the algorithm was tested for different applications, including comparing how the model performs on data preprocessed using z-norms. The graphs below show the results of the model for different values of C and K = 1. Recall that C is a tunable parameter that acts as a tradeoff between a more general model and a more specific model for the training set. If the value of C is low, the model performs well on the training set but it is too specific, while a large value of C guarantees a generalization but there could be non-optimal values of min_{DCF} From the results obtained, a good value for C could be 10 or 1, as we note from the results that in this range of values the model has minimal values of min_{DCF} and the model is still not too specific.

In the following table, we are showing some results using C = 10 and different values of prior:

RAW / $C = 10$	min_{DCF} ($\pi = 0.5$)	min_{DCF} ($\pi = 0.1$)	min_{DCF} ($\pi = 0.9$)
$\pi_T = 0.5$	0.122	0.293	0.371
$\pi_T = 0.1$	0.119	0.340	0.354
$\pi_T = 0.9$	0.123	0.305	0.369

Z-norm / $C = 10$	min_{DCF} ($\pi = 0.5$)	min_{DCF} ($\pi = 0.1$)	min_{DCF} ($\pi = 0.9$)
$\pi_T = 0.5$	0.118	0.297	0.350
$\pi_T = 0.1$	0.121	0.332	0.334
$\pi_T = 0.9$	0.119	0.300	0.366



(a) RAW and Z-norm $\pi_T = 0.5$

(b) RAW and Z-norm $\pi_T = 0.5$

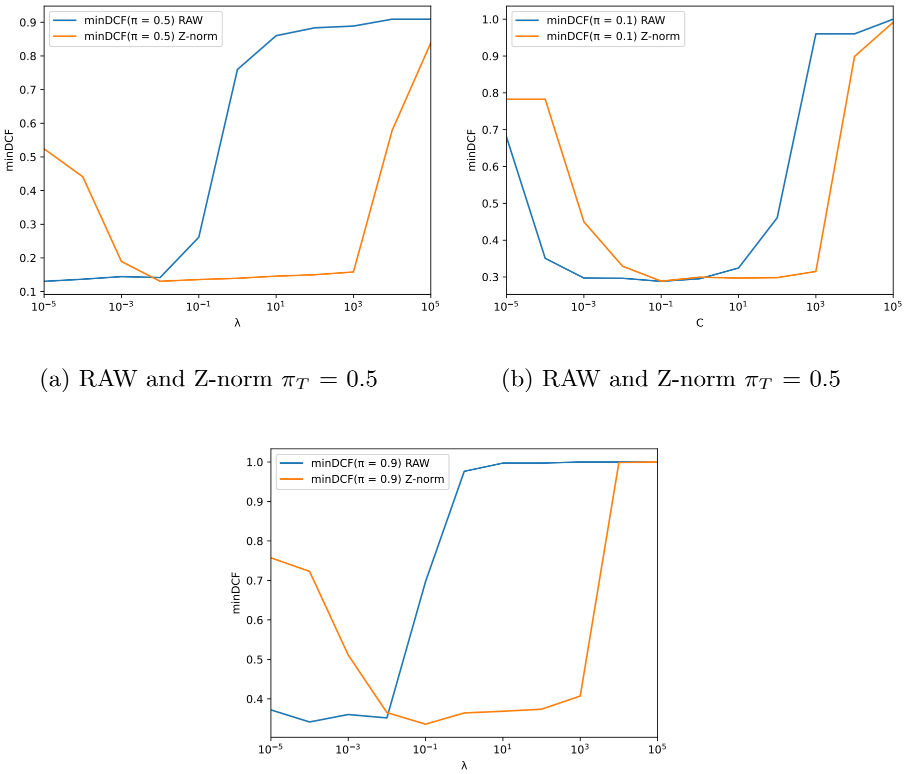
(c) RAW and Z-norm $\pi_T = 0.5$

2.4.2 Non-Linear SVM results

As a first nonlinear model we present the results obtained using Polynomial kernels using d = 2 and c = 1 as parameters. Regarding our application on RAW features, it seems that the best value of C is 10^{-3} while for features preprocessed with znorm we will choose C = 10.

RAW / $C = 10^{-3}$	min_{DCF} $(\pi = 0.5)$	min_{DCF} $(\pi = 0.1)$	min_{DCF} $(\pi = 0.9)$
$\pi_T = 0.5$	0.144	0.329	0.362
$\pi_T = 0.1$	0.145	0.367	0.338
$\pi_T = 0.9$	0.168	0.381	0.445

Znorm / $C = 10$	min_{DCF} $(\pi = 0.5)$	min_{DCF} $(\pi = 0.1)$	min_{DCF} $(\pi = 0.9)$
$\pi_T = 0.5$	0.146	0.321	0.369
$\pi_T = 0.1$	0.151	0.345	0.37
$\pi_T = 0.9$	0.151	0.380	0.428



(a) RAW and Z-norm $\pi_T = 0.5$

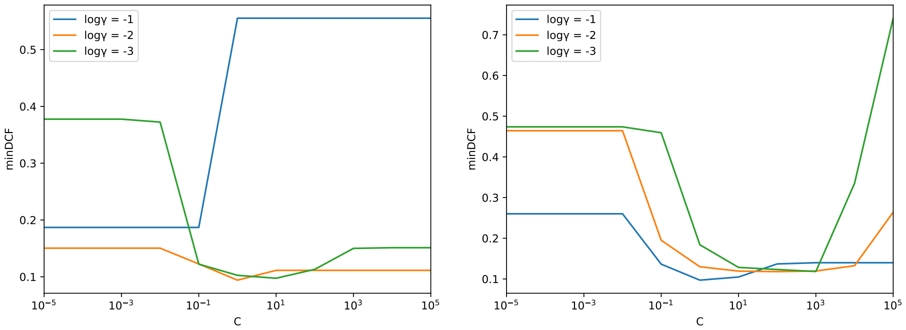
(b) RAW and Z-norm $\pi_T = 0.5$

(c) RAW and Z-norm $\pi_T = 0.5$

In following section we present results concerning the Gaussian Radial Basis Function kernel. Observing the graphs below, in which three models with different values of λ are plotted, we chose to analyze in more detail the configuration $\lambda = 0.001$ and C = 10 for RAW features and $\lambda = 0.1$ and C = 5 for znorm features.

RAW / $\gamma = 0.001 - C = 10$	min_{DCF} $(\pi = 0.5)$	min_{DCF} $(\pi = 0.1)$	min_{DCF} $(\pi = 0.9)$
$\pi_T = 0.5$	0.097	0.249	0.276
$\pi_T = 0.1$	0.106	0.356	0.241
$\pi_T = 0.9$	0.115	0.269	0.363

Znorm / $\gamma = 0.1 - C = 5$	min_{DCF} $(\pi = 0.5)$	min_{DCF} $(\pi = 0.1)$	min_{DCF} $(\pi = 0.9)$
$\pi_T = 0.5$	**0.096**	0.282	0.264
$\pi_T = 0.1$	0.105	0.362	0.235
$\pi_T = 0.9$	0.116	0.252	0.372



(a) RAW $\pi_T = 0.5$

(b) Z-norm $\pi_T = 0.5$

2.5 Gaussian Mixture Model

The last classifier we are interested in is the GMM which can be considered as an extension of the first classifier we analyzed in this project. The GMM assumes that distributions of data can be explained by the use of multiple Gaussians. This assumption is very important to us in fact from the analysis of the dataset it was found that some features seem to be constituted in just this way. The explanation we have also previously given is that the dataset contains data from three different age groups. From the results obtained with the Gaussian Model, we expect that the Standard GMM and the Tied GMM will have fairly good performance, while the Diagonal GMM and the Tied GMM will have suboptimal results due to the same considerations made earlier about the correlation of features in the dataset.

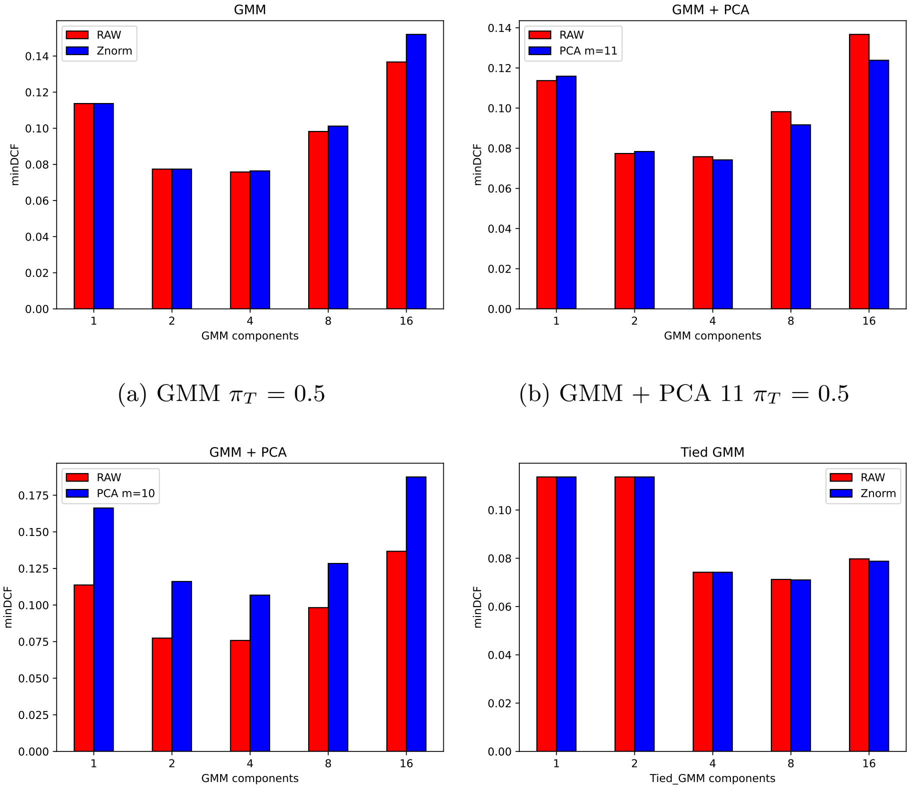
2.5.1 GMM results

As we can see from the graphs, the results obtained agree with our expectations. The best models are precisely GMM and Tied GMM just as they had been for the Gaussian Classifier. The Diagonal GMM and TiedDiagonal, on the other hand, show up as expected, significantly worse. Focusing on the first two models, we also checked whether PCA could be useful, finding that for m=11 this helps slightly to improve performance but only for some component values. Turning to the components , the same number was used for both male and female class since they had a similar distribution. As expected, the best results are obtained for component values of 2 or 4 with standard GMM for RAW features. This result can be explained precisely by the fact that our dataset showed distributions consisting of 2-3 Gaussians. Results for a number of components greater than 16 were omitted as not significant to our discussion.

Model / Components	min_{DCF} $(\pi = 0.5)$	min_{DCF} $(\pi = 0.1)$	min_{DCF} $(\pi = 0.9)$
GMM / (2)	**0.077**	0.249	0.204
GMM / (4)	**0.076**	0.227	0.214
GMM + PCA11 / (4)	**0.074**	0.215	0.209
Tied GMM / (8)	**0.071**	0.206	0.209

Model / Components	min_{DCF} ($\pi = 0.5$)	min_{DCF} ($\pi = 0.1$)	min_{DCF} ($\pi = 0.9$)
GMM + znorm / (2)	0.077	0.249	0.204
GMM + znorm/ (4)	0.076	0.227	0.216
Tied GMM + znorm / (8)	**0.071**	0.193	0.198

The top 4 models are the GMM with 4 components, GMM + PCA11 with 4 components and the Tied GMM with 8 components and the Tied GMM +znorm with 8 components. Overall, the results are very similar so we will consider as best the GMM + PCA11 with 4 components since it follows more the number of clusters we observed in the dataset.

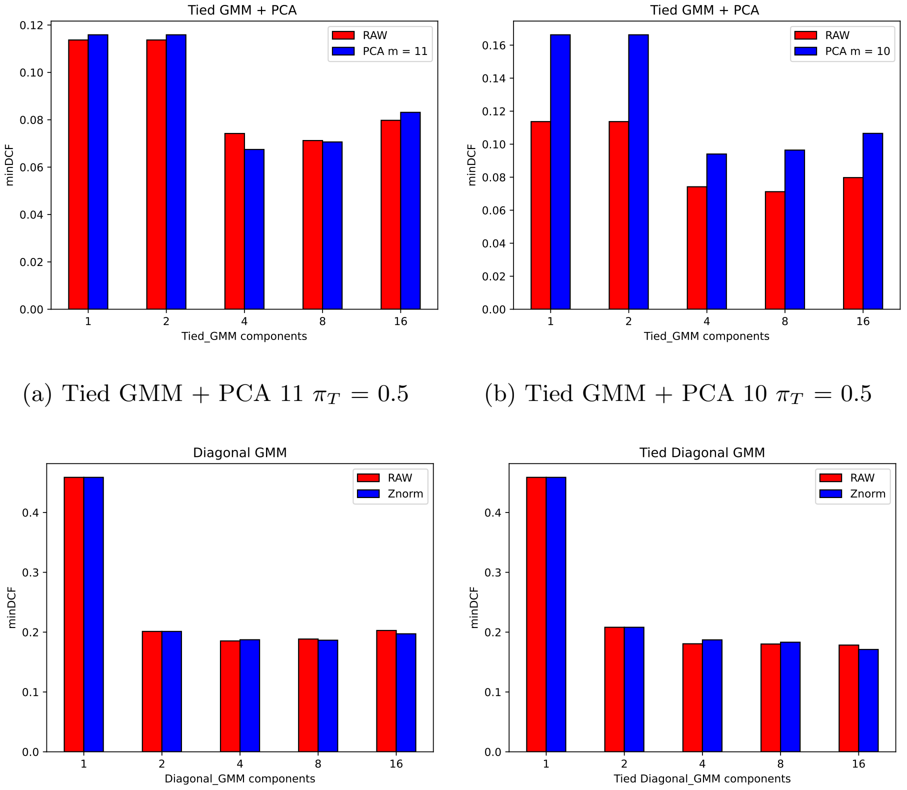


(a) GMM $\pi_T = 0.5$

(b) GMM + PCA 11 $\pi_T = 0.5$

(c) GMM + PCA 10 $\pi_T = 0.5$

(d) Tied GMM $\pi_T = 0.5$



(a) Tied GMM + PCA 11 $\pi_T = 0.5$

(b) Tied GMM + PCA 10 $\pi_T = 0.5$

(c) Diagonal GMM $\pi_T = 0.5$

(d) Tied Diagonal GMM $\pi_T = 0.5$

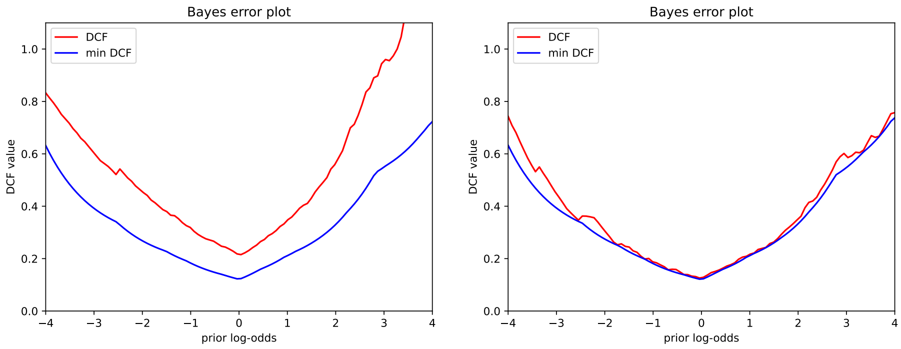
2.6 Candidate Models

To summarize what we have seen so far, we show in this table the candidate models for our evaluation. Standard MVG has been omitted since GMM seems more promising.

Model	min_{DCF} $(\pi = 0.5)$
LR ($\lambda = 0$, $\pi_T = 0.9$)	**0.117**
RKB-SVM (Z-norm, $\lambda = 0.1$, C = 5, $\pi_T = 0.5$)	**0.096**
GMM (PCA11 , 4 components)	**0.074**

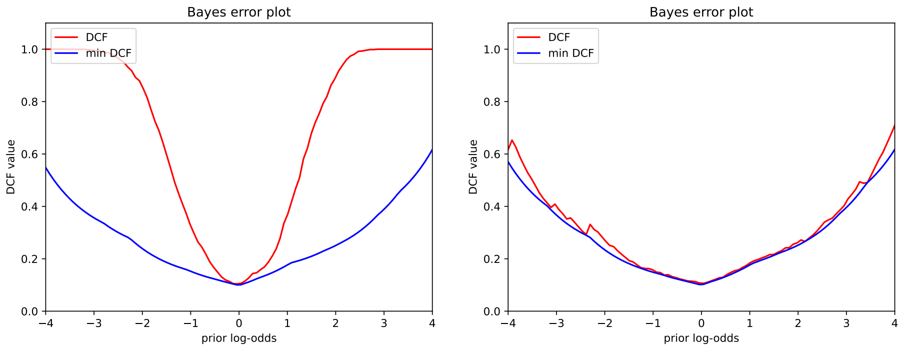
3 Calibration

To assess the performance of various models, we've thus far relied exclusively on the minimum cost of detection (minDCF), which, however, is contingent on a specific threshold. In order to determine whether the threshold aligns with the theoretical one, we'll employ a metric known as actual DCF (actDCF). The choosen calibration model for this process is based on Logistic Regression. Given our limited dataset, we'll employ a K-Fold approach to estimate the parameters of the calibration function. This approach ensures robust parameter estimation and generalization across the data.



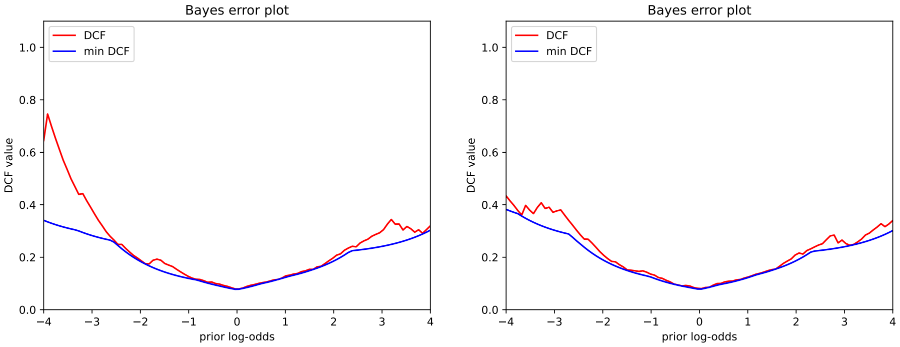
(a) LR uncalibrated

(b) LR calibrated



(a) RKB-SVM uncalibrated

(b) RKB-SVM calibrated



(a) GMM uncalibrated

(b) GMM calibrated

As we can see, LR and RKB-SVM have non calibrated scores so calibration is effective. GMM does not need calibration because it is weel calibrated yet,nevertheless we reported the calibrated results anyway. Withthese plots, it is also possible to observe how the model behaves for different applications

4 Evaluation