Gender Detection Analysis

Alessandro Mandrile

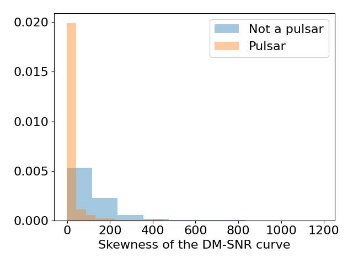
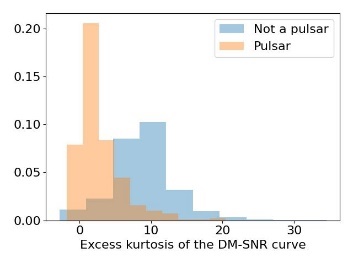
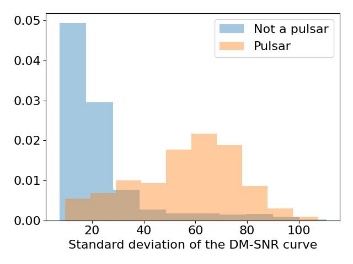
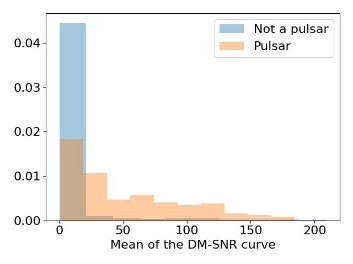
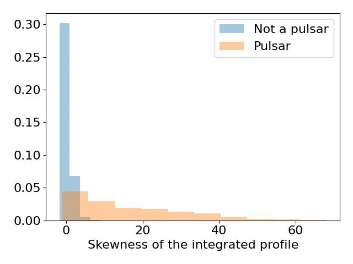
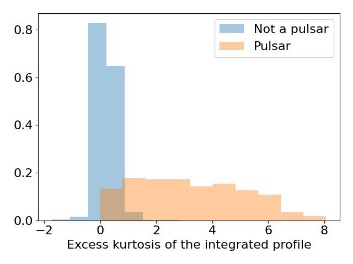
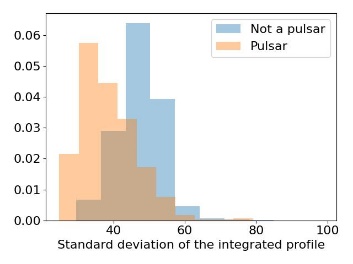
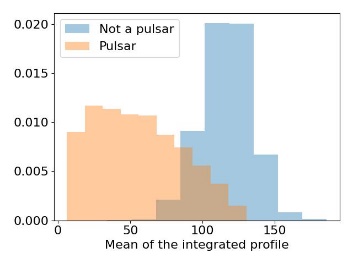
* Introduction

The dataset consists of synthetic speaker embeddings that represent the acoustic characteristics of a spoken utterance. A speaker embedding is a small-dimensional, fixed sized representation of an utterance. Features are continuous values that represent a point in the m-dimensional embedding space. The embeddings have already been computed and reduced to 12-dimensional feature space.

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 meaning. Speakers belong to four different age groups. The age information, however, is not available.

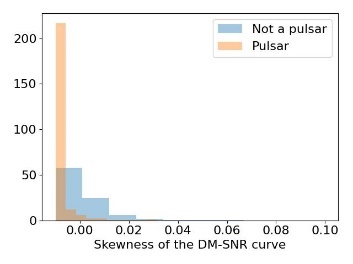
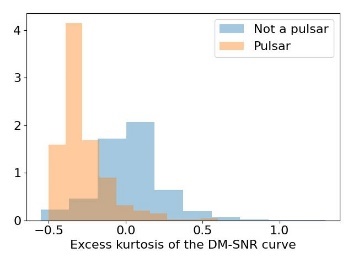
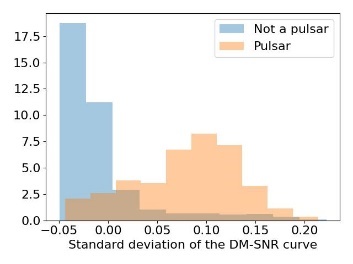
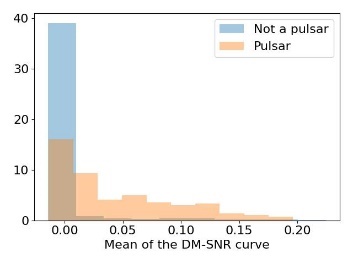
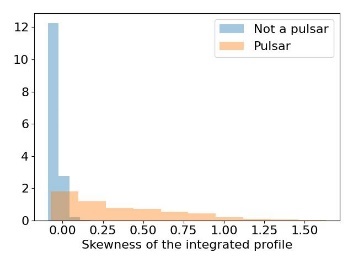
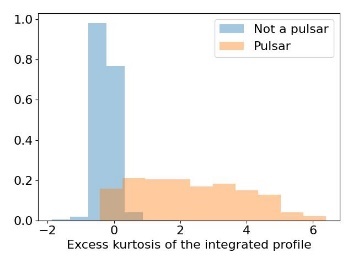
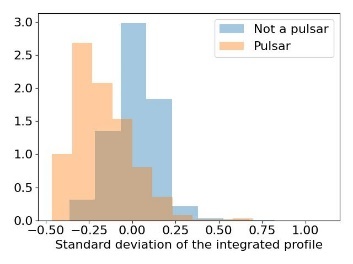
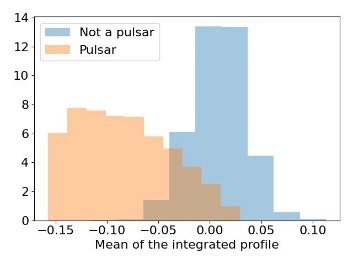
* Features

Histogram of the HTRU2 dataset features (training set). Features are sorted by their order, from left to right, top to bottom.

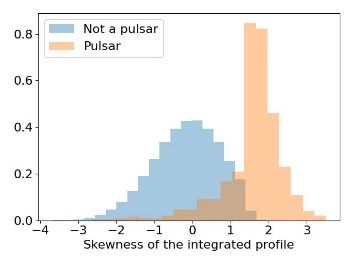
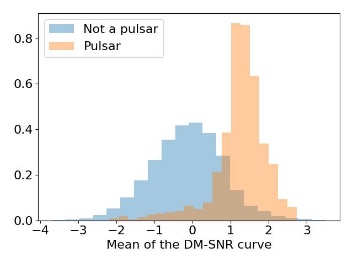
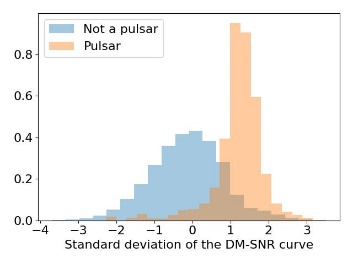
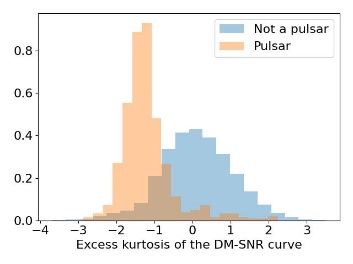


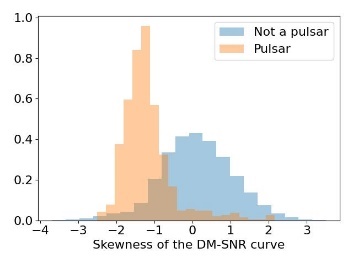
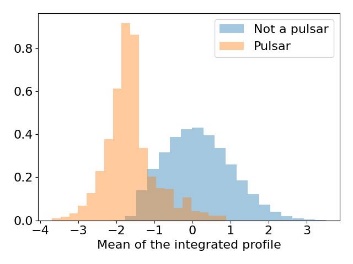
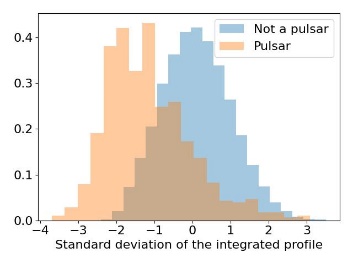
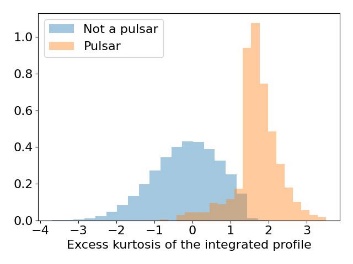
After a first analysis it’s clear that features 3, 4 and 7 are more like exponential distribution, rather than normal densities. Therefore, the whole research could have benefit from a transformation of those features to normal ones. For all the others we can see that, even for features not really close to normal densities (like 5, Non pulsar), classes are well separated.

One big problem of HTRU2 is features’ range, as values ranges from 10^0 to 10^3, so we also z-normalized the dataset. Here the histograms of z-normalized features.



We can see that now ranges are much closer, even though Excess kurtosis features now have slightly higher values. To solve exponential features problem the last pre-processing step will be a guassianification of the dataset.



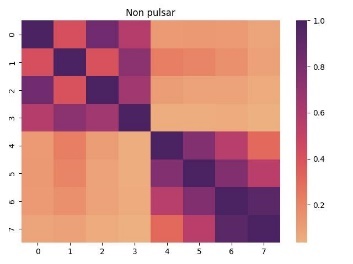
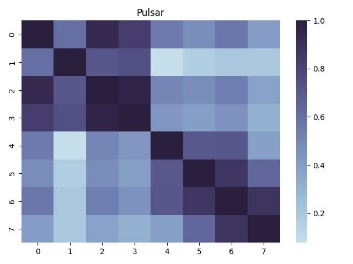
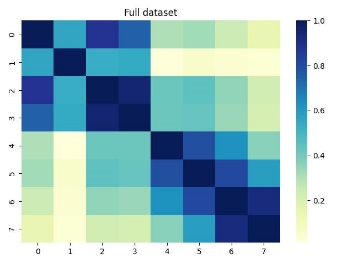


Finally, classes are well divided. This will be helpful for generative classifiers

Final notes: Z-Normalisation and Gaussianify assumes that Train data and Test data will have similar distributions, since the transformations are computed on train data and applied on test.

* Multivariate Gaussian classifier

Now we will consider Guassian classifiers, but first it is better to check the absolute value of the Pearson correlation coefficient . Darker color implies larger value



Since features 0-3 and 4-7 are extracted from the same statistics it could be expected that they would have had higher correlations. A PCA could help but it would mean reducing too much the dataset, since we would have need to do a PCA-5.

Since within class covariance are far from diagonal, we will only test full-covariance models. In the process of choosing the best model and for optimizing hyperparameter, K-fold cross-validation has been used. Since the number of train samples is high, K=5 is chosen.

The main application will be a uniform prior one: (, Cfp, Cfn) = (0.5, 1, 1), but we will also will also consider unbalanced applications(, Cfp, Cfn) = (0.1, 1, 1), since the train dataset is much more unbalanced towards that direction.

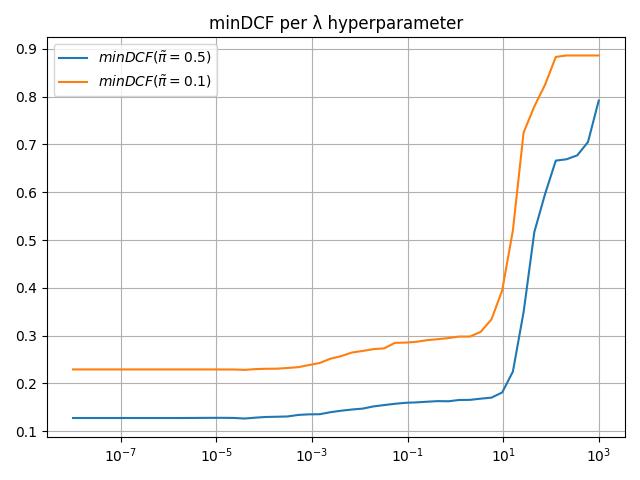
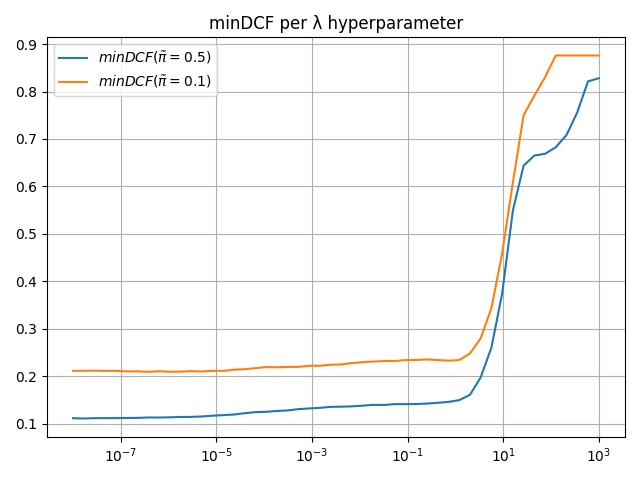
MVG classifier – min DCF on the validation set

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Raw features | | |
| Full-Cov | 0.142 | 0.278 |
| Tied-Cov | 0.110 | 0.224 |
| Guassianized features | | |
| Full-Cov | 0.158 | 0.248 |
| Tied-Cov | 0.131 | 0.231 |
| Z-normalised features | | |
| Full-Cov | 0.142 | 0.278 |
| Tied-Cov | 0.110 | 0.224 |

Results shows that Tied Covariances always wins over Full covariances. This might be because we failed in asserting the independence assumption on features. In addition, this could mean that this kind of problem is more characterised by a linear separation rule. For next evaluation we will use the MVG Tied covariances model as representative of the generative models.

* Logistic regression model

We will now focus on logistic regression model, firstly by selecting an optimal λ for unregularized dataset. Once found a good value, we will compare the effect of rebalancing the Logistic regression cost function. Here we can compare the effect of different λ on z-normalized (left) and guassianized (right) features.

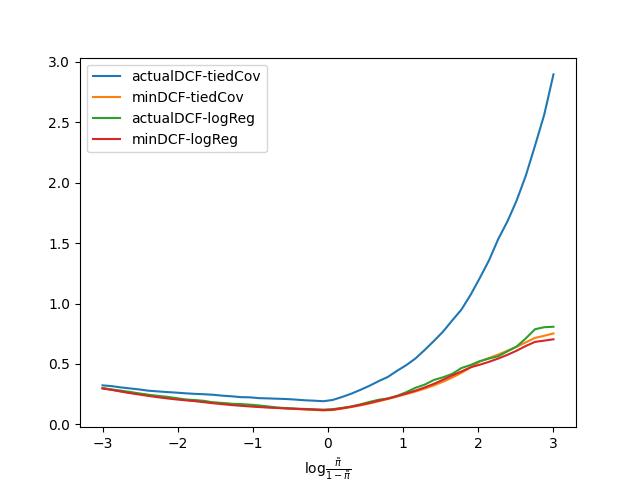


The two diagrams show that for λ lower than 10-3 gaussianification doesn’t change results. In addition, it seems like also not balanced application get good performance with this classifier. To select a proper λ value we consider two values: 10-4 and 10-5; the next table contains also different min DCF for regularized dataset and for the two different considered applications.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Raw | | |
| Log Reg (λ = 10−5, πT = 0.5) | 0.115 | 0.215 |
| Log Reg (λ = 10−5, πT = 0.1) | 0.114 | 0.213 |
| Raw | | |
| Log Reg (λ = 10−4, πT = 0.5) | 0.116 | 0.214 |
| Log Reg (λ = 10−4, πT = 0.1) | 0.112 | 0.212 |
| Z-normalized | | |
| Log Reg (λ = 10−4, πT = 0.5) | 0.122 | 0.220 |
| Log Reg (λ = 10−4, πT = 0.1) | 0.125 | 0.219 |
| Guassianized | | |
| Log Reg (λ = 10−4, πT = 0.5) | 0.125 | 0.236 |
| Log Reg (λ = 10−4, πT = 0.1) | 0.130 | 0.235 |

In red are highlighted the overall best values for a regularized dataset. We cans see that a lower λ doesn’t give a big boost, while if we go too low on it, we might overfit our data. Therefore, we select 10-4. In blue it is also highlighted how a πT closer to our dataset empirical prior probability even improve scores. However, for now we select the regularized version, which will help for the evaluation of the best threshold, but we can keep in mind that in case the train set would have much different priors from the train dataset, we will just drop the regularization.

Here we can see a Bayes error plot with normalized minimum Detection Cost Function, to show how much and where the Logistic Regression model performs with respect to MVG.

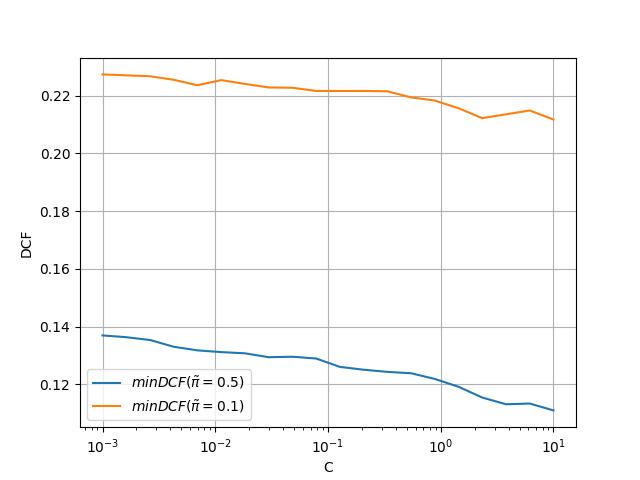


These results encourage the idea that separation surfaces are linear, in fact the two models behave almost equally, despite the logistic Regression having an already optimal threshold (embedded in regularization of the objective function). Tied covariances, instead, needs some more tweaking. Finally, application seems to handle better low , a behaviour that we will not change since it is the kind of priors we expect while recognising Pulsar (rare celestial object).

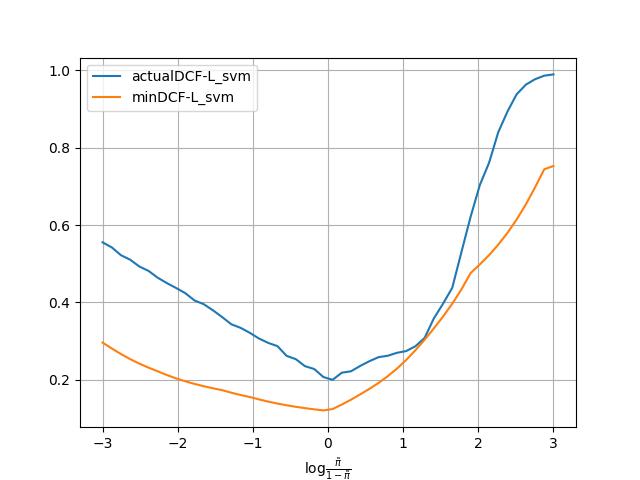
After these results we will focus only on models providing linear separation surfaces.

* Support Vector Machine

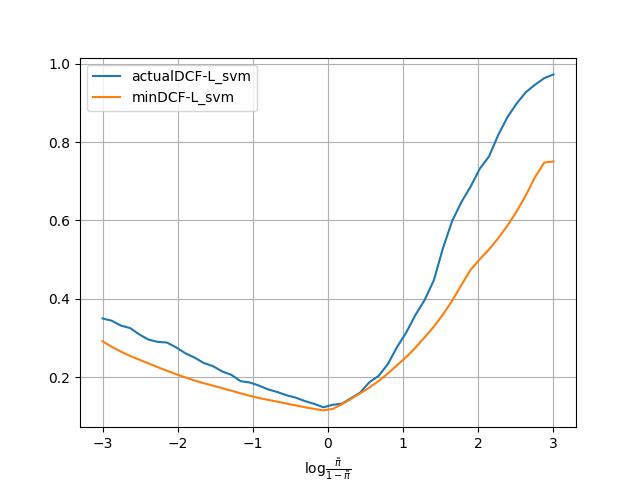
The main hyperparameter to tune for a linear kernel support vector machine is the C upper limit for αi values. Since for high values of C the Dual problem to minimize becomes computationally too intensive, we looked for the lower value giving an almost good minDCF. Having already found models with a minDCF of 0.111 we chose the first value under 0.12, so C=2, but knowing that one the other hyper-parameters are tuned, we can still increase a bit C at cost of time. The graph below has been obtained with K-Fold (K=4) for C from 10-3 to 101.



Here an error Bayes plot for linear SVM, C = 2 and unregularized C proportions (meaning πT= πemp). MinDCF is like logistic regression and Tied MVG, but actual DCF is much worse. All evaluation still done on the training set, through K-Mean cross validation.



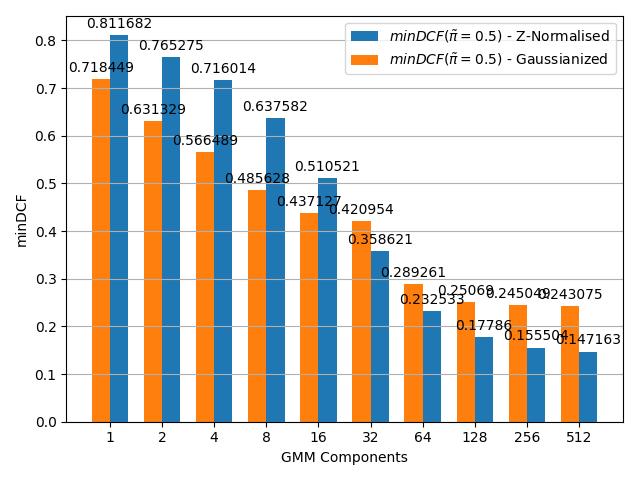
We will now rebalance the Dual problem C bounds and plot the actual DCF but keep K at 1 since tuning also K takes too much time.



We will skip non linear kernel for the Support vector machine, since we already saw the best results are obtained for linear separation surfaces.

* Gaussian Mixture Model

The last generative model is GMM, for which we will only see the Tied Covariance version, following what seen for MVG. Unfortunately, the EM algorithm we have is bugged (log-likelihood ratios increase at some EM iterations), so decisions taken by this model are sub-optimal. Anyway, a GMM trained with the actual algorithm got minDCF close the best linear model for G greater than 256. We will show the results, but not use the model.

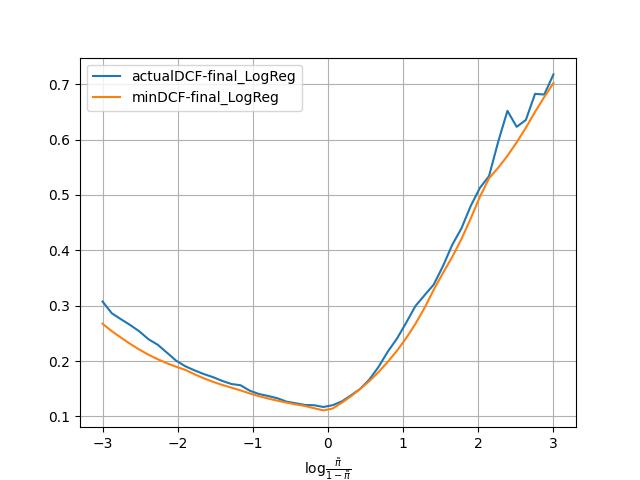


We will choose Logistic Regression because, even though has small differences from the other model, it is already well calibrated and ready to use.

So, our last proposal is Logistic Regression with λ = 10−4 and balanced class. While still keeping an eye on Tied MVG. A possible second solution could also have been a fusion of the two models, maybe with a score calibration

* Experimental result

We applied the chosen model on the test data, in raw, and here the results. As expected the DCF are good.



The actual DCF is 0.114 on the Test set, showing that HRTU2 dataset can be classified by linear classification rules.