Pulsar Detection: Classification of HTRU2 dataset

Salvatore Licata
Email: s295798@studenti.polito.it

***Abstract*—The goal of this analysis is to exploit some of the most common Machine Learning (ML) algorithms in order to classify the HTRU2 dataset. This dataset is composed of data collected by observing almost 18 thousand astronomical objects, aiming to identify pulsars. We will start analyzing the features, understanding their distributions and correlation, and then we will move to the analysis of the performance of different classifiers, followed by score calibration of the best ones. The results will lead us to choose quadratic models, which proved to be the most effective ones.**

I. Introduction

HTRU2 is a data set which describes a sample of pulsar candidates collected during the High Time Resolution Universe Survey (South).

Pulsars are a rare type of Neutron star that produce radio emission detectable here on Earth. They are of considerable scientific interest as probes of space-time, the inter- stellar medium, and states of matter.

As pulsars rotate, their emission beam sweeps across the sky, and when this crosses our line of sight, produces a detectable pattern of broadband radio emission. As pulsars rotate rapidly, this pattern repeats periodically. Thus pulsar search involves looking for periodic radio signals with large radio telescopes.

Each pulsar produces a slightly different emission pattern, which varies slightly with each rotation. Thus a potential signal detection known as a 'candidate', is averaged over many rotations of the pulsar, as determined by the length of an observation. In the absence of additional info, each candidate could potentially describe a real pulsar. However in practice almost all detections are caused by radio frequency interference (RFI) and noise, making legitimate signals hard to find.

Machine learning tools are now being used to automatically label pulsar candidates to facilitate rapid analysis. Classification systems in particular are being widely adopted, which treat the candidate data sets as binary classification problems. Here the legitimate pulsar examples are a minority positive class, and spurious examples the majority negative class.

The data set shared here contains 16,259 spurious examples caused by RFI/noise, and 1,639 real pulsar examples. These examples have all been checked by human annotators. Each candidate is described by 8 continuous variables. The dataset is split into Train and Evaluation data. The training set contains 8108 false pulsar signal and 821 true pulsar signal, while the evaluation set contains 8151 false pulsar signal and 818 true pulsar signal. Candidates are stored in files in separate rows. Each row lists the eight features plus a label: 0 if false pulsar, 1 if true pulsar. We will classify the proposed task using different models, in order to compare the results and choosing best candidates.

II. Features Analysis

We will now focus on analyzing the features of the train dataset. Each candidate is described by 8 continuous variables and a single binary class variable. The features are listed below:

1) Mean of the integrated profile.
2) Standard deviation of the integrated profile.
3) Excess kurtosis of the integrated profile.
4) Skewness of the integrated profile.
5) Mean of the DM-SNR curve.
6) Standard deviation of the DM-SNR curve.
7) Excess kurtosis of the DM-SNR curve.
8) Skewness of the DM-SNR curve.

In order to avoid computation problems, data were pre-processed using Z-normalization:

$$z_i = \frac{x_i - \mu}{\sigma}, \forall i \in \{1, ..., n\} \tag{1}$$

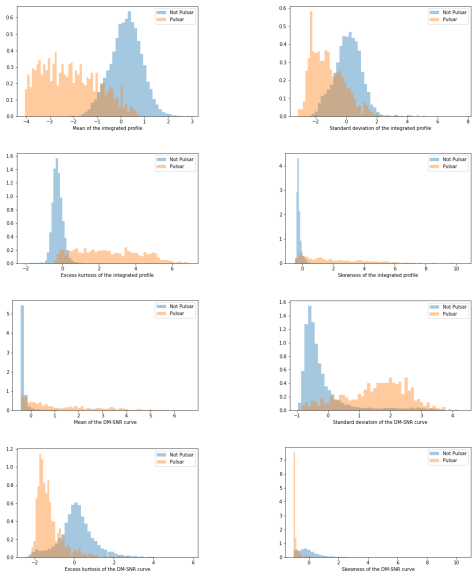

Figure 1: Features after z-normalization (blue false pulsar, red true pulsar)

We can notice that features seem already well distributed. For the sake of completeness, we report here also their gaussianized version.

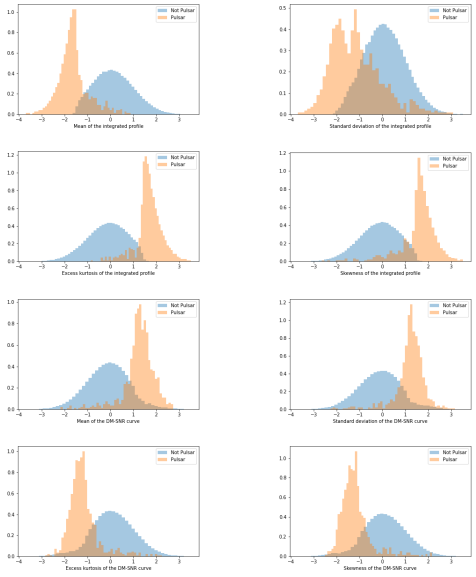

Figure 2: Features after gaussianization (blue false pulsar, red true pulsar)

We can now proceed to analyze the correlation coefficient of the features. We decided to use Pearson's correlation coefficient:

$$r = \frac{Cov(X,Y)}{\sqrt{Var(X)}\sqrt{Var(Y)}} \tag{2}$$

Below there are the reported results divided in three groups: considering both classes, considering only false pulsars and considering only true pulsar.

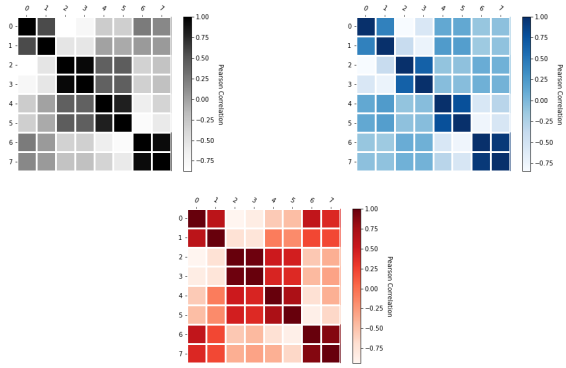

Figure 3: Pearson correlation coefficient (grey both classes, blue false pulsar, red true pulsar)

We can notice that features 2,3 and 6,7 are strongly correlated, while 4,5 are moderately correlated. This suggests us that employing a dimensionality reduction technique like Principal Component Analysis (PCA) could allow us to reduce the size of the samples without major losses of information. Candidates values for PCA are 7, 6 and 5 (even if we expect higher loss of information with the latter).

III. Classification Models

We will perform our analysis employing K-folds cross-validation, with K = 3. We will consider three applications: a balanced one and two unbalanced ones which bias towards one of the classes respectively.

$$\begin{aligned}(\tilde{\pi}, C_{fn}, C_{fp}) &= (0.5, 1, 1)\\ (\tilde{\pi}, C_{fn}, C_{fp}) &= (0.1, 1, 1)\\ (\tilde{\pi}, C_{fn}, C_{fp}) &= (0.9, 1, 1)\end{aligned} \tag{3}$$

Our target application will be the balanced one. We want to find the most promising approach, so we will measure performances using normalized minimum Detection Cost Function (DCF), a metric to measure the cost we would pay for making optimal decisions using scores by a recognizer R. The performances will be evaluated over a validation subset extracted from the training set (one for each fold).

IV. Gaussian Classifiers

We start considering the main Multi-Variate Gaussian (MVG) classifiers: Full Covariance, Diagonal Covariance, Tied Covariance and Tied Diagonal Covariance. All of them are generative models which assume data follow a gaussian distribution, given the class:

$$X|C \sim \mathcal{N}(\mu_c, \sigma_c) \tag{4}$$

The Tied MVG classifier assumes a shared covariance matrix for all classes, the Diagonal one assumes the covariance matrices are diagonal (i.e. features are independently distributed so there is no correlation between them), while the Tied Diagonal one is a combination of the previous two, so there is a shared diagonal covariance matrix. Following the analysis on the features, we expect that the best models will be the Full Covariance MVG and the Tied Covariance MVG, since features are not uncorrelated, leading the Naive Bayes assumption (which allows us to diagonalize covariance matrices) to larger errors.

Min DCF using 3-folds validation

$\tilde{\pi}$:	0.5	0.1	0.9
MVG Full:	0.143	0.659	0.289
MVG Diag:	0.192	0.734	0.316
MVG Tied:	0.112	0.569	0.225
MVG Tied Diag:	0.160	0.580	0.266

Table I: Z-Normalized features no PCA

Min DCF using 3-folds validation

$\tilde{\pi}$:	0.5	0.1	0.9
MVG Full:	0.139	0.625	0.307
MVG Diag:	0.214	0.718	0.509
MVG Tied:	0.112	0.565	0.224
MVG Tied Diag:	0.139	0.594	0.273

Table II: Z-Normalized features PCA 7

Min DCF using 3-folds validation

$\tilde{\pi}$:	0.5	0.1	0.9
MVG Full:	0.152	0.635	0.292
MVG Diag:	0.222	0.727	0.534
MVG Tied:	0.138	0.573	0.255
MVG Tied Diag:	0.163	0.597	0.302

Table III: Z-Normalized features PCA 6

Min DCF using 3-folds validation

$\tilde{\pi}$:	0.5	0.1	0.9
MVG Full:	0.150	0.647	0.254
MVG Diag:	0.220	0.749	0.459
MVG Tied:	0.149	0.569	0.263
MVG Tied Diag:	0.17	0.586	0.313

Table IV: Z-Normalized features PCA 5

Min DCF using 3-folds validation

$\tilde{\pi}$:	0.5	0.1	0.9
MVG Full:	0.154	0.696	0.242
MVG Diag:	0.154	0.603	0.279
MVG Tied:	0.133	0.534	0.234
MVG Tied Diag::	0.163	0.607	0.295

Table V: Gaussianized features no PCA

Min DCF using 3-folds validation

$\tilde{\pi}$:	0.5	0.1	0.9
MVG Full:	0.153	0.697	0.243
MVG Diag:	0.167	0.659	0.248
MVG Tied:	0.135	0.546	0.241
MVG Tied Diag::	0.137	0.568	0.256

Table VI: Gaussianized features PCA 7

Min DCF using 3-folds validation

$\tilde{\pi}$:	0.5	0.1	0.9
MVG Full:	0.155	0.687	0.242
MVG Diag:	0.159	0.64	0.241
MVG Tied:	0.137	0.541	0.247
MVG Tied Diag::	0.142	0.589	0.259

Table VII: Gaussianized features PCA 6

Min DCF using 3-folds validation

$\tilde{\pi}$:	0.5	0.1	0.9
MVG Full:	0.151	0.691	0.249
MVG Diag:	0.157	0.642	0.246
MVG Tied:	0.137	0.534	0.248
MVG Tied Diag::	0.143	0.591	0.257

Table VIII: Gaussianized features PCA 5

Analyzing the results, we can deduce that gaussianization is not necessary and worsens performances (we had already noticed that the features were well distributed). PCA with m=7 seems to perform best, while reducing more dimensions seems to cause information losses. Thus, we will analyze the following classifiers taking into account z-normalized features only and PCA with m = 7 / no PCA. For the MVG classifiers, our top-performer is the MVG with Tied Covariance applied on Z-normalized features with PCA (m = 7).

V. Logistic Regression Classifiers

We will now start to consider discriminative model. Logistic regression is an example of this class. We can employ two types of Logistic Regression: Linear and Quadratic (through Feature Expansion). Moreover, since classes are not balanced (the majority of data belongs to the false pulsar class) we will use a prior weighted version of Logistic Regression, see Equation 5.

$$J(w,b) = \frac{\lambda}{2}\|w\| + \frac{\pi_T}{n_T}\sum_{i=1|c_i=1}^{n} \log(1+e^{-z_i(w^T x+b)}) + \\ + \frac{1-\pi_T}{n_F}\sum_{i=1|c_i=0}^{n} \log(1+e^{-z_i(w^T x+b)}) \tag{5}$$

The model parameters are w and b, while λ is an hyperparameter which provides regularization. In the following analysis we will use $\pi_t = 0.5$ because we want to target our main application. We will start analyzing the Linear type, trying to estimate the best hyperparameter (λ).

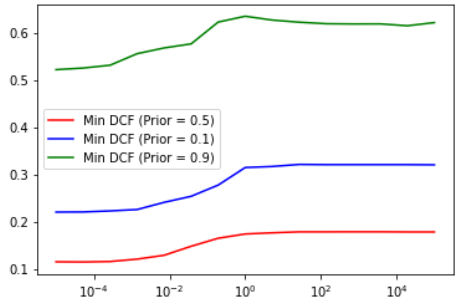

Figure 4: Min DCFs for different values of λ and different priors

We can see from the graph that we obtain best results when λ is smaller. For the classification part we chose to use $\lambda = 10^{-5}$. Following, there are the results for the Linear Logistic Regression Classifier:

Min DCF using 3-folds validation

$\tilde{\pi}$:	0.5	0.1	0.9
Linear LR ($\pi_T = 0.5$, $\lambda = 10^{-5}$):	0.116	0.221	0.523

Table IX: Z-Normalized features no PCA

Min DCF using 3-folds validation

$\tilde{\pi}$:	0.5	0.1	0.9
Linear LR ($\pi_T = 0.5$, $\lambda = 10^{-5}$):	0.116	0.220	0.536

Table X: Z-Normalized features PCA 7

Once again we can notice that results using PCA with m = 7 do not show significant losses of data. Linear Logistic Regression seems to achieve good results, so now we will concentrate on exploiting feature expansion in order to obtain a Quadratic Logistic Regression, to see if it can obtain even better results. Feature Expansion is a technique to extend the dimensionality of a sample x, allowing us to create a mapping $\mathbb{R}^n \rightarrow \mathbb{R}^m$ where $m > n$. In our case, since we are interested in quadratic expansion, we can obtain this mapping by using as samples an "extended version" through Equation 6.

$$\phi(x) = \begin{bmatrix} vec(xx^T) \\ x \end{bmatrix} \qquad w' = \begin{bmatrix} vec(A) \\ b \end{bmatrix}, \quad b' = c \tag{6}$$

The operator $vec(M)$ is the operator that stacks the columns of matrix M. Thanks to the new feature space, we can obtain a modified version of the Logistic Regression equation, leading us to:

$$J(w', b') = \frac{\lambda}{2}\|w'\| + \frac{\pi_T}{n_T}\sum_{i=1|c_i=1}^{n} \log(1 + e^{-z_i(w'^T\phi(x)+b')}) + \\ + \frac{1-\pi_T}{n_F}\sum_{i=1|c_i=0}^{n} \log(1 + e^{-z_i(w'^T\phi(x)+b')}) \tag{7}$$

The model parameters are still w and b, so we will have to estimate again the best hyperparameter (λ).

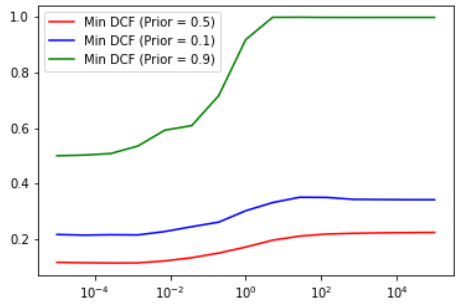

Figure 5: Min DCFs for different values of λ and different priors

We can see from the graph that we obtain again best results when λ is smaller. For the classification part we chose to use $\lambda = 10^{-5}$. Following, there are the results for the Quadratic Logistic Regression Classifier:

Min DCF using 3-folds validation

$\tilde{\pi}$:	0.5	0.1	0.9
Quadratic LR ($\pi_T = 0.5$, $\lambda = 10^{-5}$):	0.118	0.218	0.501

Table XI: Z-Normalized features no PCA

Min DCF using 3-folds validation

$\tilde{\pi}$:	0.5	0.1	0.9
Quadratic LR ($\pi_T = 0.5$, $\lambda = 10^{-5}$):	0.115	0.218	0.500

Table XII: Z-Normalized features PCA 7

From the results achieved we can conclude that Linear and Quadratic Logistic Regression classifiers achieve similar results, with the latter being slightly better in case of unbalanced applications. We will then choose as best model for Logistic Regression the Quadratic one, with $\pi_T = 0.5$, $\lambda = 10^{-5}$.

VI. Support Vector Machine Classifiers

The risk minimization approach seen in Logistic Regression, through the Regularization parameter λ, allowed us to avoid growing w indefinitely, in case of linearly separable data. The risk minimization problem can be cast in a more general way, allowing us to separate data up to a margin, giving a geometrical definition to the regularization term, identified by the so called Support Vectors. This strategy leads to a new type of classifiers, hence called Support Vector Machine (SVM) models. Moreover, SVM classifiers, in contrast with Logistic Regression ones, do not provide probabilistic scores which, then, cannot be interpreted as class posterior.

The formulation of SVM can be cast into two equations, called the Primal and the Dual formulation, see Eq. 8 and 9.

$$\hat{J}^P(w, b) = \frac{1}{2}\|w\| + C\sum_{i=1}^{n}\max[1 - z_i(w^T x + b)]_+ \tag{8}$$

$$\hat{J}^D(\alpha) = \alpha^T 1 - \frac{1}{2}\alpha^T \hat{H}\alpha \\ \hat{H}_{ij} = z_i z_j \hat{x}_i^T \hat{x}_j \\ \hat{x} = \begin{bmatrix} x \\ K \end{bmatrix} \\ \text{where } 0 \leq \alpha_i \leq C_i, \forall i \in \{1, ..., n\} \tag{9}$$

Since the dual formulation is easier to optimize and it allows us to produce non-linear hyperplanes without computing feature expansion (it simply requires that we compute dot products in the expanded space, through a kernel function), we will exploit it and we will not use the Primal one. Specifically we will use $K = 1$ and we will balance classes using different bounds for α, changing C_i according to the sample class, see Eq. 10.

$$C_i = \begin{cases} C_T = C\frac{\pi_T}{\pi_T^{emp}}, & \text{if i } \in \mathcal{H}_T \\ C_F = C\frac{\pi_F}{\pi_F^{emp}}, & \text{if i } \in \mathcal{H}_F \end{cases} \tag{10}$$

In the following we will analyze three types of SVM: Linear, Quadratic and Radial Basis Function (RBF) SVMs. Linear SVMs have only one hyperparameter, which is C. Following, a graph comparing the variation of the min DCF according to different C values and priors.

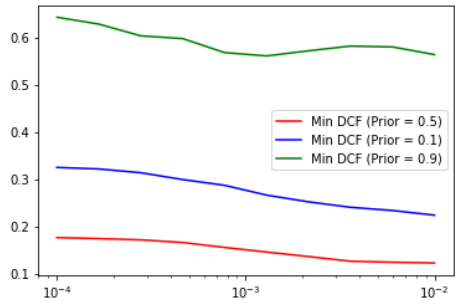

Figure 6: Min DCFs for different values of C and different priors

We can thus choose $C = 0.01$, since it seems to guarantee a lover min DCF. In the following we show the results obtained on z-normalized features without PCA and with PCA (m = 7).

Min DCF using 3-folds validation

$\tilde{\pi}$:	0.5	0.1	0.9
Linear SVM ($\pi_T = 0.5$, C = 0.01):	0.124	0.224	0.563

Table XIII: Z-Normalized features no PCA

Min DCF using 3-folds validation

$\tilde{\pi}$:	0.5	0.1	0.9
Linear SVM ($\pi_T = 0.5$, C = 0.01):	0.123	0.225	0.565

Table XIV: Z-Normalized features PCA 7

We can confirm again the PCA with m = 7 is not detrimental, however, the results are slightly worse than the previous classifiers. We will now turn our attention to Non-Linear SVMs. Since the dual formulation simply depends on the dot product of the features, see Eq. 9, we can find a kernel function $k(\hat{x}_i, \hat{x}_j) = \phi(\hat{x}_i)^T \phi(\hat{x}_j)$ which allows us to compute the dot product as if we were in an expanded feature space. We can define the following kernel functions:

$$k(\hat{x}_i, \hat{x}_j) = (\hat{x}_i^T \hat{x}_j + c)^d$$
$$k(\hat{x}_i, \hat{x}_j) = e^{-\gamma \|\hat{x}_i - \hat{x}_j\|} \tag{11}$$

We will employ the first equation with $d = 2, c = 1$ for quadratic SVM, while we will perform a grid search to find optimal γ (also called the width of the kernel) with respect to C for RBF SVM.

Following, a graph comparing the variation of the min DCF according to different C values and priors.

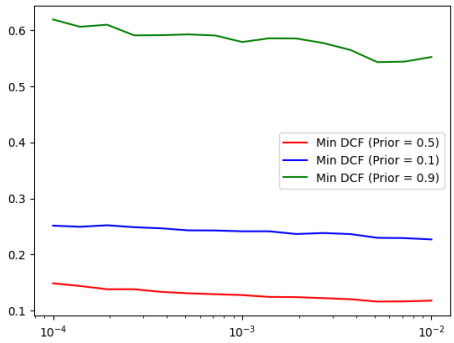

Figure 7: Min DCFs for different values of C and different priors

We can thus choose again $C = 0.01$, since it seems to guarantee a lover min DCF. In the following we show the results obtained on z-normalized features without PCA and with PCA (m = 7).

Min DCF using 3-folds validation

$\tilde{\pi}$:	0.5	0.1	0.9
Quadratic SVM ($\pi_T = 0.5$, C = 0.01):	0.118	0.227	0.552

Table XV: Z-Normalized features no PCA

Min DCF using 3-folds validation

$\tilde{\pi}$:	0.5	0.1	0.9
Quadratic SVM ($\pi_T = 0.5$, C = 0.01):	0.118	0.227	0.549

Table XVI: Z-Normalized features PCA 7

We can see again that the results using PCA with m = 7 are good enough, in this case also better than without PCA. At the moment, quadratic SVM seems to be the best SVM model, obtaining results closer to the best between other classifiers. To complete the SVM classifiers, we now turn our attention to RBF SVM. Following there is a comparison between the minDCF obtained with different width of the kernel (γ) and different values of C.

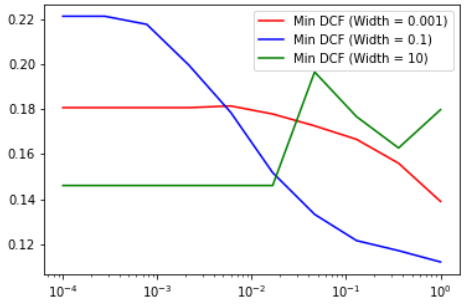

Figure 8: Min DCFs for different values of C and different priors

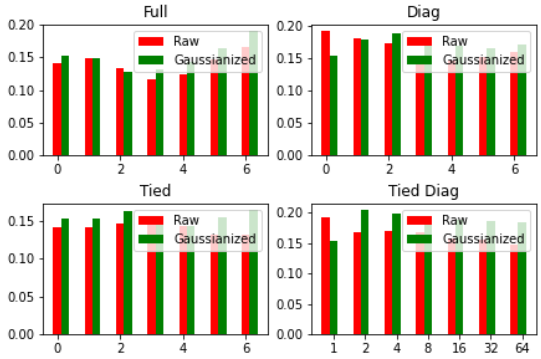

Figure 9: Min DCFs for different values of components

According to the results we can choose $C = 0.01, \gamma = 0.1$, since they seem to give the best results. We train our model once again on z-normalized features without and with PCA (m = 7).

Min DCF using 3-folds validation

$\tilde{\pi}$:	0.5	0.1	0.9
RBF SVM ($\pi_T = 0.5$, C = 1, $\gamma = 0.1$):	0.112	0.219	0.535

Table XVII: Z-Normalized features no PCA

Min DCF using 3-folds validation

$\tilde{\pi}$:	0.5	0.1	0.9
RBF SVM ($\pi_T = 0.5$, C = 1, $\gamma = 0.1$):	0.114	0.219	0.540

Table XVIII: Z-Normalized features PCA 7

The RBF SVM seems to obtain slightly better results than the quadratic one, so we can conclude that our best SVM model is the RBF SVM, with $C = 1, \gamma = 0.1$ applied on Z-normalized features with PCA (m = 7).

VII. Gaussian Mixture Classifiers

The last classifier we will consider is a Gaussian Mixture Model (GMM) classifier. It is a generative classifier, which assumes (like MVG) that data are gaussian distributed. However, differently from the MVG, the GMM supposes that for each class the distribution ca be modelled by a collection of gaussian distribution. The number of gaussian components is an hyperparameter of the model. Below a comparison of the minDCF variation with different number of components for z-normalized and gaussianized data and for different types of GMMs (Full Covariance, Diagonal Covariance, Tied Covariance and Tied Diagonal Covariance).

It is worth to be mentioned that the Tied version is different from the concept of Tied MVG, because the covariance matrix is shared not between classes, but between different components of the same class. Considering Figure 9, we can identify 4 candidates models, all based on z-normalized features (since gaussianization once again does not seem to help): Full Covariance GMM with 8 components, Diagonal Covariance GMM with 16 components, Tied Covariance and Tied Diagonal Covariance GMM with 64 components. Below we can find the results:

Min DCF using 3-folds validation

$\tilde{\pi}$:	0.5	0.1	0.9
GMM Full (8 gau):	0.115	0.231	0.552
GMM Diag (16 gau):	0.146	0.29	0.663
GMM Tied (64 gau):	0.131	0.24	0.686
GMM Tied Diag (64 gau):	0.148	0.291	0.663

Table XIX: Z-Normalized features no PCA

Min DCF using 3-folds validation

$\tilde{\pi}$:	0.5	0.1	0.9
GMM Full (8 gau):	0.125	0.229	0.601
GMM Diag (16 gau):	0.136	0.227	0.502
GMM Tied (64 gau):	0.134	0.246	0.695
GMM Tied Diag (64 gau):	0.131	0.245	0.595

Table XX: Z-Normalized features PCA 7

Analyzing the results, we can see that the best candidate seems to be the Full Covariance GMM with 8 components, applied on z-normalized features without PCA. This concludes the classification part, now we will take into consideration the best models overall and we will apply score calibration on them, to assess their performances before proceeding to the experimental results.

VIII. Scores Calibration

Summarizing the previous results, we selected a candidate model from each type of classifier:

1) MVG with Tied Covariance applied on Z-normalized features with PCA (m = 7)

2) Quadratic Logistic Regression ($\pi_T = 0.5$, $\lambda = 10^{-5}$) applied on Z-normalized features with PCA (m = 7)
3) RBF SVM ($\pi_T = 0.5$, C = 1, $\gamma = 0.1$) applied on Z-normalized features with PCA (m = 7)
4) Full Covariance GMM with 8 components, applied on z-normalized features without PCA

However, up to now we considered only the min DCF metric, which allowed to measure the cost we would pay if we made optimal decisions for the evaluation set using the recognizer scores The cost that we actually pay, however, depends on the goodness of the threshold we use to perform class assignment, therefore we now turn our attention to actual DCFs.

Following, we can see the Bayes Error Plot (BEP) for different applications for each of the selected models.

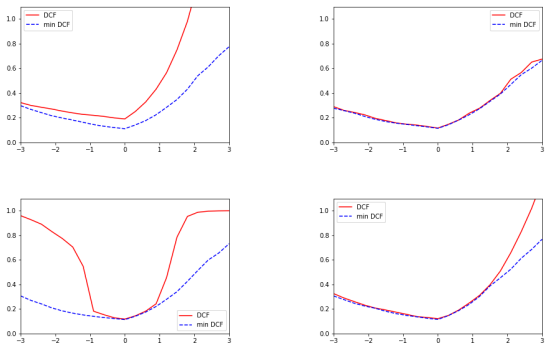

Figure 10: Min DCF vs Actual DCF, from Top to Bottom, from Left to Right: Tied MVG, Quadratic LR, Quadratic SVM, Full Cov GMM

As we can see, only the Quadratic Logistic Regression classifier seems to produce already calibrated scores (which is not a surprise, since we employed a prior weighted version), thus we can resort to score calibration for the other three models. In the following, we will perform score calibration using a Prior Weighteed Linear Logistic Regression, which starting from the score s, it is able to estimate a function $f(s) = \alpha s + \beta$. Since $f(s)$ should produce well-calibrated scores, it can be interpreted as the log-likelihood ratio for the two class hypothesis:

$$f(s) = \frac{f_{S|C}(s|\mathcal{H}_T}{f_{S|C}(s|\mathcal{H}_F} = \alpha s + \beta \tag{12}$$

The class posterior probability for prior $\tilde{\pi}$ corresponds to:

$$\log \frac{P(C = \mathcal{H}_T|s}{P(C = \mathcal{H}_F|s} = \alpha s + \beta + \log \frac{\tilde{\pi}}{1 - \tilde{\pi}} = \alpha + \beta' \tag{13}$$

Thus, we can finally see that we can recover calibrated scores using a Weighted Linear Logistic Regression Model by the following:

$$f(s) = \alpha s + \beta' - \log \frac{\tilde{\pi}}{1 - \tilde{\pi}} \tag{14}$$

Notice that we will calibrate scores for a specific application ($\tilde{\pi} = 0.5$, but we will see that other applications will benefit of the calibration too. In the following you can see the comparison of min DCFs versus actual DCFs after we calibrated scores (for all models except Quadratic LR, which was producing already calibrated scores).

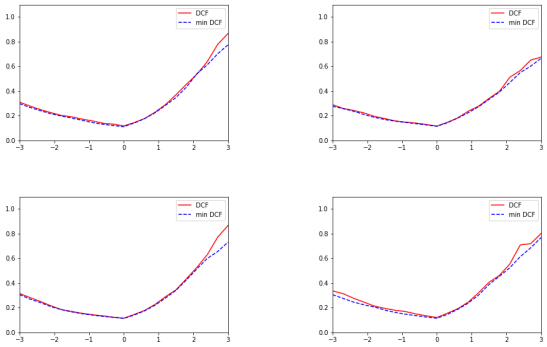

Figure 11: Min DCF vs Actual DCF after calibration, from Top to Bottom, from Left to Right: Tied MVG, Quadratic LR, Quadratic SVM, Full Cov GMM

We can notice that, as we said, even if we calibrated scores for a specific application, also other applications see an improvement. In particular, the RBF is the model which could benefit the most from the calibration process. Following there are also more specific tables reporting the comparison of min DCF vs Actaul DCF for our three applications only (before and after calibration).

Min DCF vs Actual DCF using 3-folds validation

$\tilde{\pi}$:	0.5		0.1		0.9	
MVG Tied:	0.112	0.191	0.224	0.274	0.565	1.422
Quadratic LR:	0.115	0.117	0.218	0.231	0.500	0.537
RBF SVM:	0.114	0.119	0.219	0.853	0.54	0.992
GMM Full Cov:	0.115	0.122	0.231	0.244	0.552	0.693

Table XXI: Z-Normalized features no PCA

Min DCF vs Actual DCF (cal) using 3-folds validation

$\tilde{\pi}$:	0.5		0.1		0.9	
MVG Tied:	0.112	0.118	0.224	0.234	0.565	0.556
Quadratic LR:	0.115	0.117	0.218	0.231	0.500	0.537
RBF SVM :	0.114	0.116	0.219	0.231	0.54	0.53
GMM Full Cov:	0.115	0.122	0.231	0.254	0.552	0.608

Table XXII: Z-Normalized features no PCA

Now we can finally apply the results to the Test dataset.

IX. Experimental Results

We now analyze the different models performances on the test set in terms of min DCFs. The results are summarized in Fig. XXIII and XXIV.

Min DCF on evaluation set

$\tilde{\pi}$:	0.5	0.1	0.9
MVG Full:	0.14	0.283	0.638
MVG Diag:	0.187	0.325	0.616
MVG Tied:	0.11	0.206	0.586
MVG Tied Diag:	0.15	0.26	0.549
Linear LR ($\pi_T = 0.5, \lambda = 1e-05$):	0.108	0.199	0.545
Quadratic LR ($\pi_T = 0.5, \lambda = 1e-05$):	0.107	0.208	0.423
Linear SVM ($\pi_T = 0.5, C = 0.01, K = 1$):	0.113	0.209	0.558
Quadratic SVM ($\pi_T = 0.5, C = 0.01$):	0.103	0.203	0.465
RBF SVM ($\pi_T = 0.5, C = 1.0, \gamma = 0.1$):	0.101	0.205	0.454
GMM Full (8 gau):	0.109	0.223	0.563
GMM Diag (16 gau):	0.134	0.259	0.593
GMM Tied (64 gau):	0.128	0.24	0.58
GMM Tied Diag (64 gau):	0.139	0.286	0.617

Table XXIII: Z-Normalized features no PCA

Min DCF on evaluation set

$\tilde{\pi}$:	0.5	0.1	0.9
MVG Full:	0.139	0.296	0.589
MVG Diag:	0.197	0.506	0.766
MVG Tied:	0.109	0.206	0.587
MVG Tied Diag:	0.138	0.254	0.55
Linear LR ($\pi_T = 0.5, \lambda = 1e-05$):	0.106	0.201	0.549
Quadratic LR ($\pi_T = 0.5, \lambda = 1e-05$):	0.104	0.203	0.449
Linear SVM ($\pi_T = 0.5, C = 0.01, K = 1$):	0.113	0.211	0.563
Quadratic SVM ($\pi_T = 0.5, C = 0.01$):	0.104	0.203	0.47
RBF SVM ($\pi_T = 0.5, C = 1.0, \gamma = 0.1$):	0.1	0.204	0.456
GMM Full (8 gau):	0.117	0.234	0.525
GMM Diag (16 gau):	0.131	0.246	0.606
GMM Tied (64 gau):	0.136	0.235	0.624
GMM Tied Diag (64 gau):	0.144	0.257	0.574

Table XXIV: Z-Normalized features PCA (m = 7)

The results are consistent with the ones obtained in the previous analysis and the best models for our target application are still MVG Tied Cov PCA with m = 7, Quadratic Weighted Logistic Reression PCA with m = 7, RBF SVM PCA with m = 7 and GMM Full-Cov 8 GMM components no PCA. The results are similar between validation and evaluation, this suggests there are no relevant differences between training and evaluation data.

We can also show the calibrated results for the best models and a comparison of the models using Receiver operating characteristic (ROC) and Detection Error Tradeoff (DET) curves.

Min DCF vs Actual DCF on evaluation set

$\tilde{\pi}$:	0.5		0.1		0.9	
MVG Tied:	0.109	0.19	0.206	0.269	0.587	1.229
Quadratic LR:	0.104	0.108	0.203	0.206	0.449	0.463
RBF SVM:	0.1	0.104	0.204	0.827	0.456	0.988
GMM Full Cov:	0.109	0.111	0.223	0.229	0.563	0.657

Table XXV: Z-Normalized features no PCA

Min DCF vs Actual DCF (cal) on evaluation set

$\tilde{\pi}$:	0.5		0.1		0.9	
MVG Tied:	0.109	0.11	0.206	0.216	0.587	0.604
Quadratic LR:	0.104	0.108	0.203	0.206	0.449	0.463
RBF SVM:	0.1	0.102	0.204	0.208	0.456	0.476
GMM Full Cov:	0.109	0.11	0.223	0.229	0.563	0.617

Table XXVI: Z-Normalized features no PCA

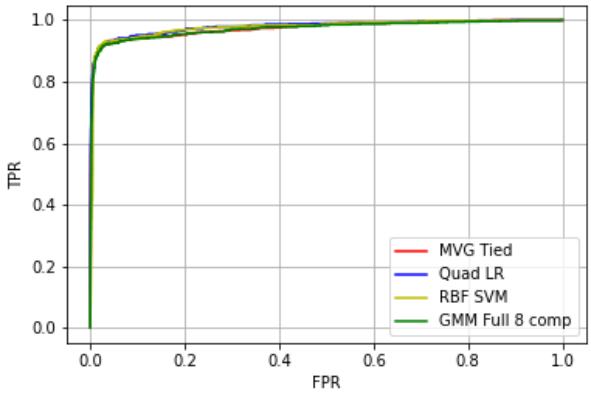

Figure 12: ROC curve comparing best models

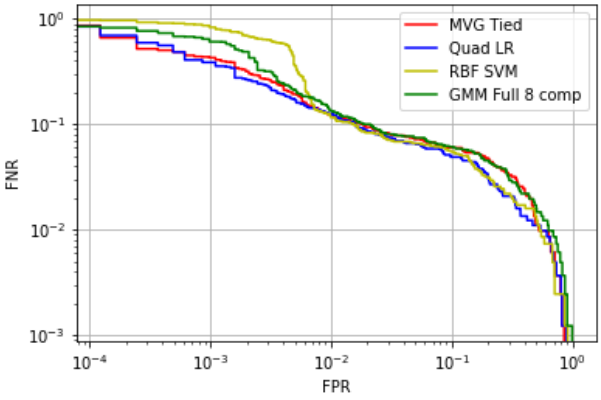

Figure 13: DET curve comparing best models

X. Conclusions

Our analysis led us to find a set of well performing models, where the Quadratic Prior Weighted Logistic Regression proved to be the most accurate, over a wide range of applications, producing also well calibrated scores. We were able to achieve a DCF of ≈ 0.1 for the target application $\tilde{\pi} = 0.5$, but also for unbalanced applications we obtained good results overall (DCF ≈ 0.2 for $\tilde{\pi} = 0.1$ and DCF ≈ 0.45 for $\tilde{\pi} = 0.9$). The choices we made on our training and validation sets proved to be correct and effective also for the evaluation set.

XI. Bibliography

- R. J. Lyon. Htru2. https://doi.org/10.6084/m9.figshare.3080389.v1
- R. J. Lyon, B. W. Stappers, S. Cooper, J. M. Brooke, J. D. Knowles, Fifty Years of Pulsar Candidate Selection: From simple filters to a new principled real-time classification approach MNRAS, 2016