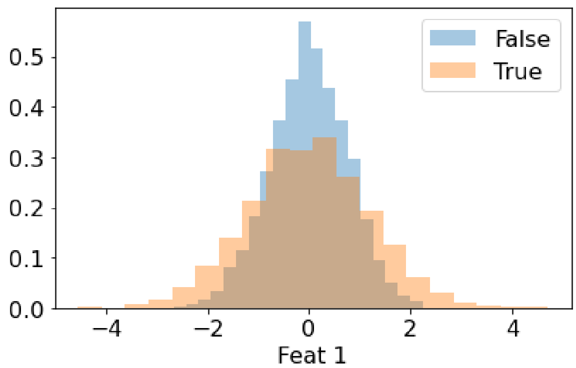
Lombardi Lorenzo Report

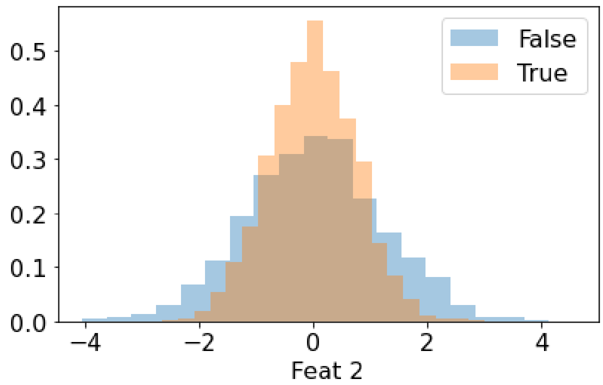
LAB2

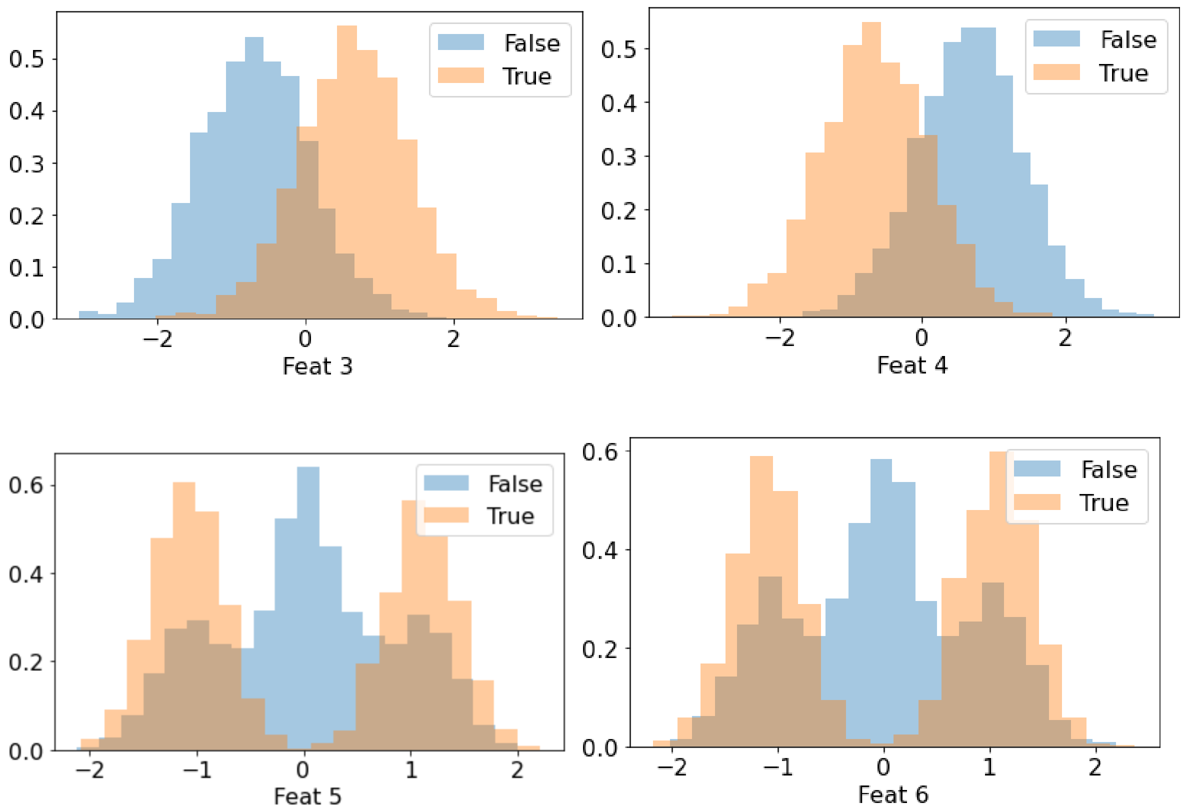
1) The classes overlap for the majority of the scatter, the “True” class is contained on the feat 2 axis but more spread apart on the feat 1 axis, while the “False” class is equally spread along the axises. The “False” class overlaps with almost the entire “True” class, but we notice a difference since “False” is more spread apart on the feat 2 axis.
The means of the first feature are respectively [0.00287744] for the False class and [5.44547838e-04] for the True, for the second feature are [0.01869316] for the False class and [-8.52437392e-03] for the True class. The means are quite similar between the two features.
The variance is respectively 0.56958105 and 0.57827792 for the first feature, while it’s 1.43023345 and 1.42086571 for the second feature, so we can notice a difference.
In feat1 there is an evident peak at 0 for the False class. The True class doesn’t have a significant peak, but there is more presence around 0.
In feat2 both classes have a peak around 0 but the True class has a more pronounced peak.

2) The classes overlap only for a portion of their data. The false class tends to have more positive values in feat 4 and more negative values in feat 3, while the true class does the opposite. The means are (respectively feat3 and feat4) [-0.68094016] [0.6708362] for False and [6.65237846e-01] [-6.64195349e-01] for True, so the means are pretty different between the two classes. Variance is 0.54997702 0.53604266 for False while it’s 0.5489026 0.55334275 for True, so this time the variance is pretty similar. We can notice peaks for feat 3 in the histogram in -1 for False and +1 for True, while for feat 4 it’s the opposite.

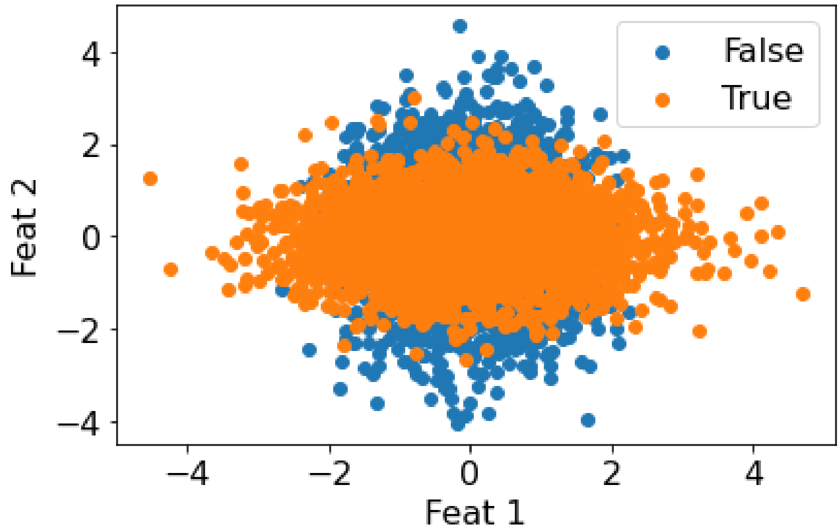
3) The scatter graph is quite interesting for feat 5 and feat 6. Each class seems to have 4 clusters of data, and only the external parts of each portions overlap with the ones of the other class. In total we can tell there are 8 clusters. The True class peaks at around 1 and -1 for both features, while the False class peaks at 0 for both features.



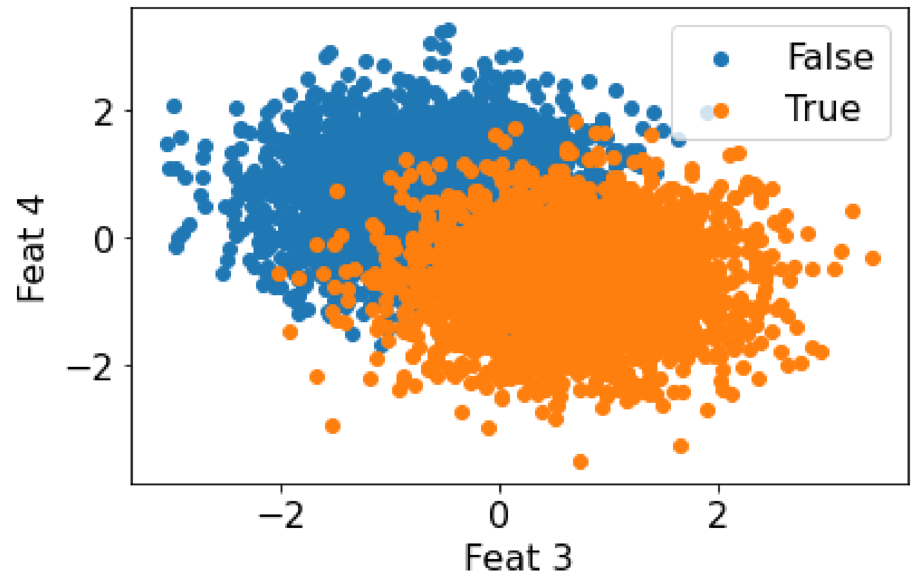


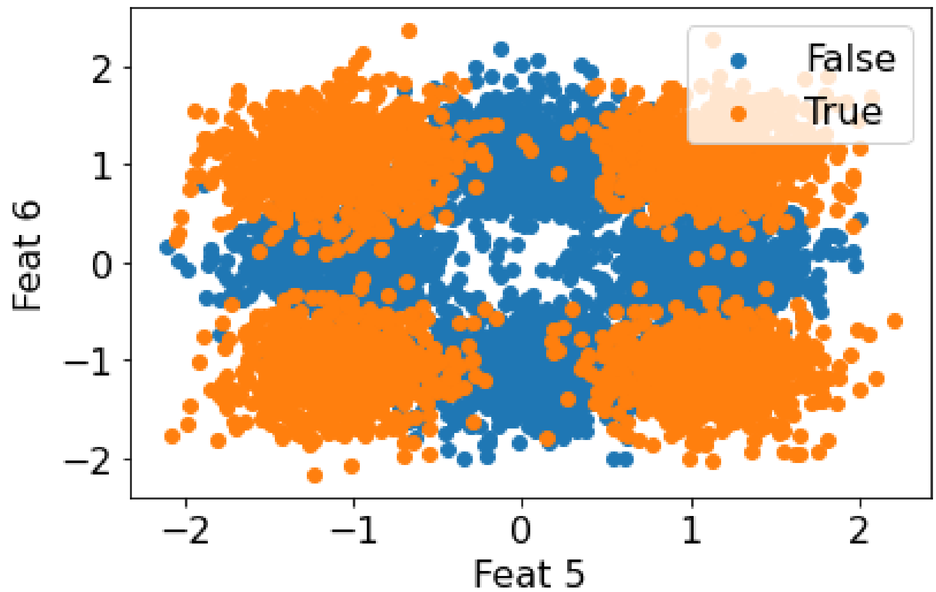


False
True
Feat 3
Feat 4
Feat 5
Feat 6



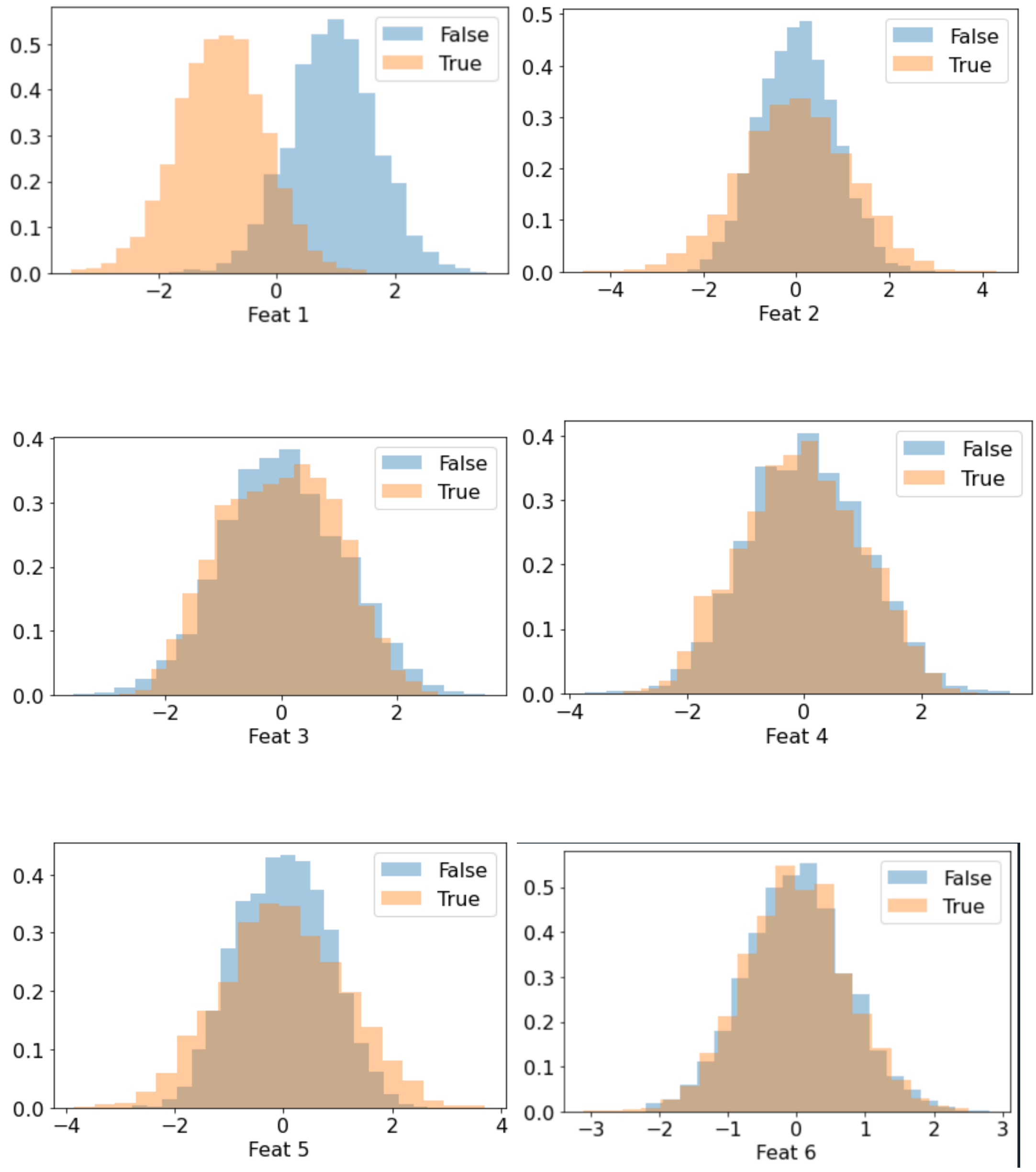
False
True
Feat 1
Feat 2



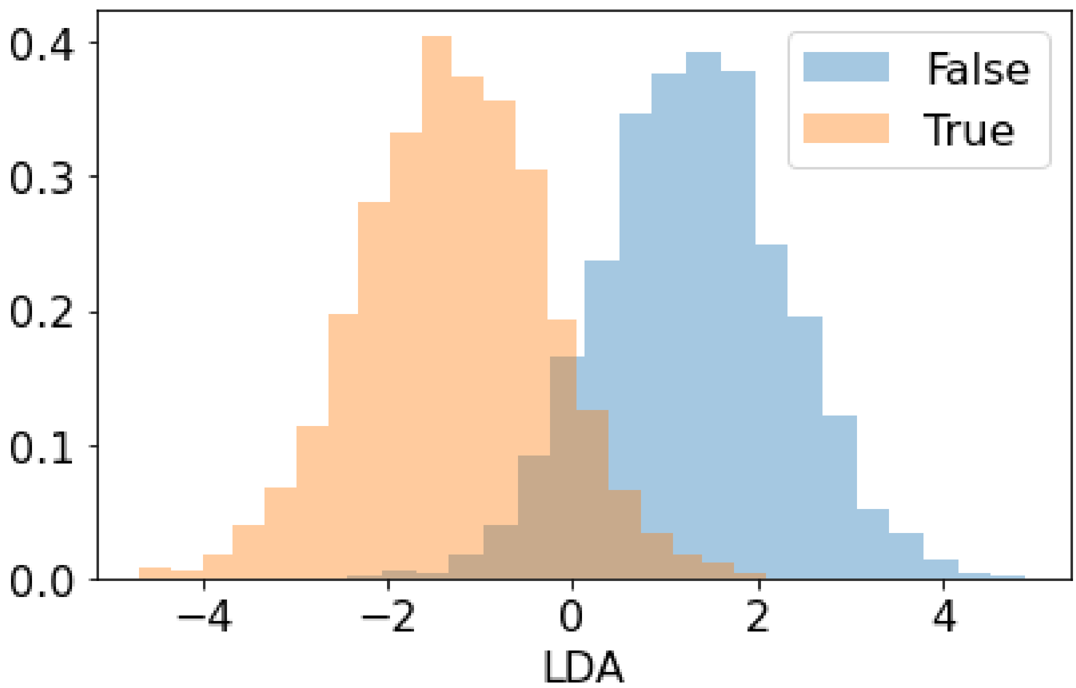


Lab3

1) From feature 2 to feature 6 we can see that most of the data is overlapping. Both classes peak around the zero mark in these features. Feature 1 presents a more interesting result, producing an histogram which doesn' have much overlapping data, with different peaks for each class. We can notice different outcomes considering what we had in Lab2, since we have all the features except 1 having lots of overlapping data. The distribution looks quite similar between every feature from 2 to 6, with both classes peaking at 0.
Clusters might appear overlapped with the exception of feat1



2) Using LDA for one dimension we can notice an overlap between -2 and +2 . The direction taken by LDA seems to be plausible with reasonable overlap of data.



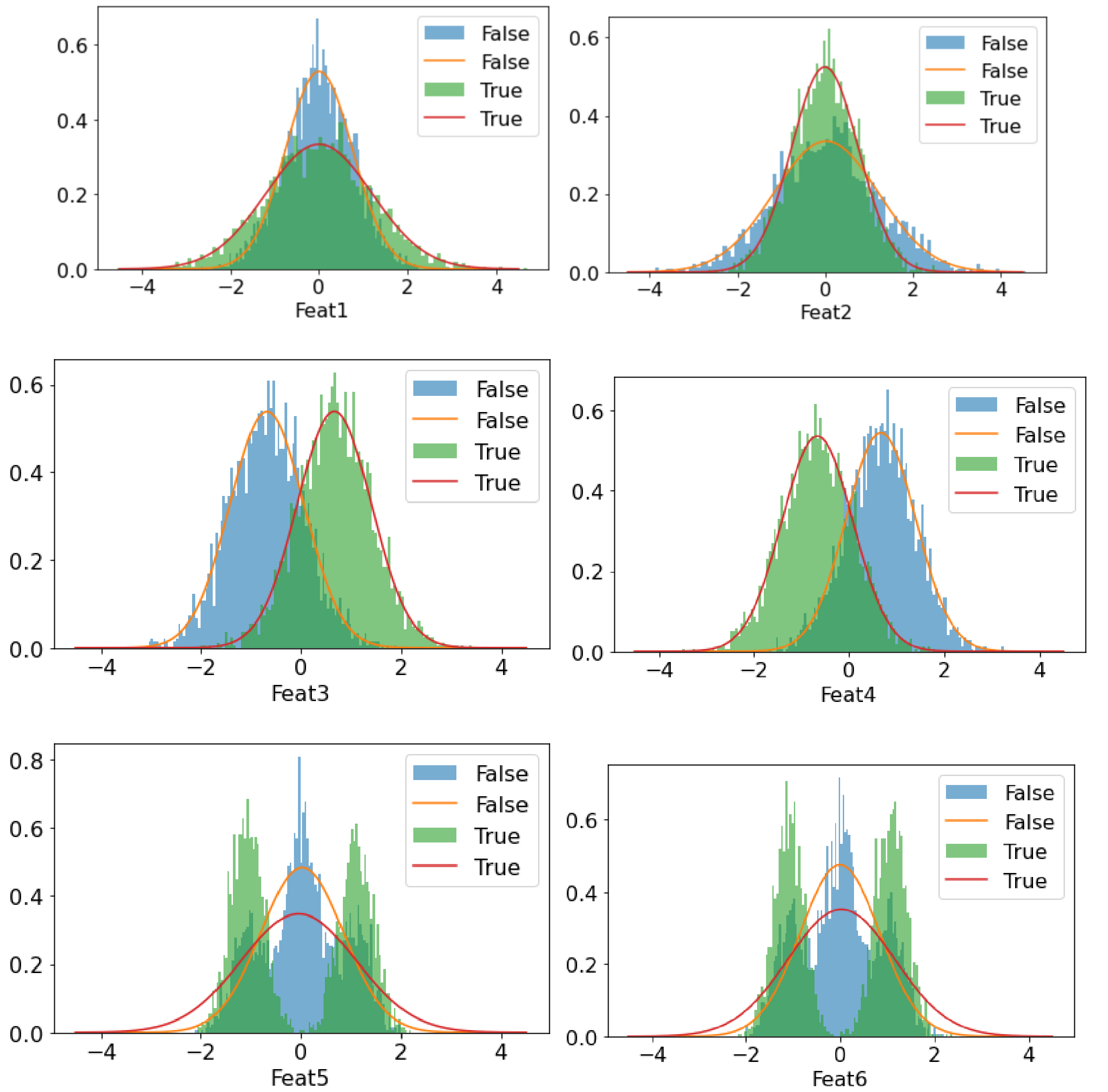
3) With one dimension, the threshold given by the means and only applying LDA , we can observe an **error rate of 9.3%.** Using a threshold too different from the original increases error rate (threshold 1= error rate 19.2%), but we can notice an improvement of 0.01% using **a threshold of 0.01.**

   Using PCA before LDA seems to be beneficial using optimal values (PCA m=2, LDA m=1) achieving an error rate of 9.2% without changing the threshold.

```
LDA
Treshold:   0.01
Labels:       [0 0 1 ... 0 0 0]
Predictions: [0 0 1 ... 0 0 0]
Number of erros: 185 (out of 2000 samples)
Error rate: 9.2%
PCA and LDA
Treshold:   -0.01828175332678561
Labels:       [0 0 1 ... 0 0 0]
Predictions: [0 0 1 ... 0 0 0]
Number of erros: 185 (out of 2000 samples)
Error rate: 9.2%
```

LAB 4

1) We can notice a good fit for the gaussian curve for Features 1,2,3,4, while for Features 5 and 6 the gaussian model struggles.



LAB5

1) **MVG - Error rate: 7.0%**
   Naive Bayes - Error rate: 7.2%
   Tied - Error rate: 9.3%
   MVG and Naive Bayes have a better rate than the 9.2% that we obtained with LDA/PCA and LDA, while Bayes Tied is closely related to LDA in terms of results
2) Covariances seem to be smaller than variances.
   Observing the covariance matrixes, we can say that the features seem weakly correlated, since the off-diagonal elements in both matrixes present values that are close to zero. This means that a Naive Bayes approach could be effective (as results show), since we consider each value indipendent from the others.
3) As said in Lab 4, the Gaussian assumptions works well for the first 4 features, but seems to struggle with feature 5 and 6.

4) Removed last 2 features:
   MVG - Error rate: 8.0%
   Naive Bayes - Error rate: 7.6%
   Tied - Error rate: 9.5%
   If we try to remove the last 2 features, every approach is slightly worse, so we can say that the last 2 features still provide useful info to some extent.
5) Features 1-2:
   MVG - Error rate: 36.5%
   Naive Bayes - Error rate: 36.3%
   Tied - Error rate: 49.5%

   We can notice pretty terrible results for every classifier. This means that feature 1-2 alone are not capable of being used on their own.

   Features 3-4:
   MVG - Error rate: 9.4%
   Naive Bayes - Error rate: 9.4%
   Tied - Error rate: 9.4%

   For features 3-4 we notice a slight worsening but not nearly as much as for features 1-2

6) **PCA MVG Error rate: 8.8%**
   PCA Naive Bayes - Error rate: 8.8%
   PCA Tied - Error rate: 9.2%
   After some trial and error for PCA i found the best m to be 2. PCA slightly worsens the MVG and Naive Bayes error rates, but doesn't seem to affect the Tied as much.

   The best accuracy has been found with **MVG used with the base classification Error Rate=7.0%**

LAB7

Prior 0.1 - Cfn 1 - Cfp 1
[[988 271]
[4 737]]

Prior 0.5 - Cfn 1 - Cfp 9
[[988 271]
[4 737]]

We can take these two results as an example. From these two matrixes we can notice that the assumption that the question asks is true. A high cost for false positive is directly related to a lower prior. The matrixes produced by the 0.1 prior with neutral costs and by the neutral prior with 9 cost for false positive present identical values.

Running experiment with MVG (No PCA)
Prior 0.5 - Cfn 1 - Cfp 1
[[927 75]
[65 933]]

DCF (non-normalized): 0.070
DCF (normalized): 0.140
MinDCF (normalized, fast): 0.130 (@ th = -2.233450e-01)

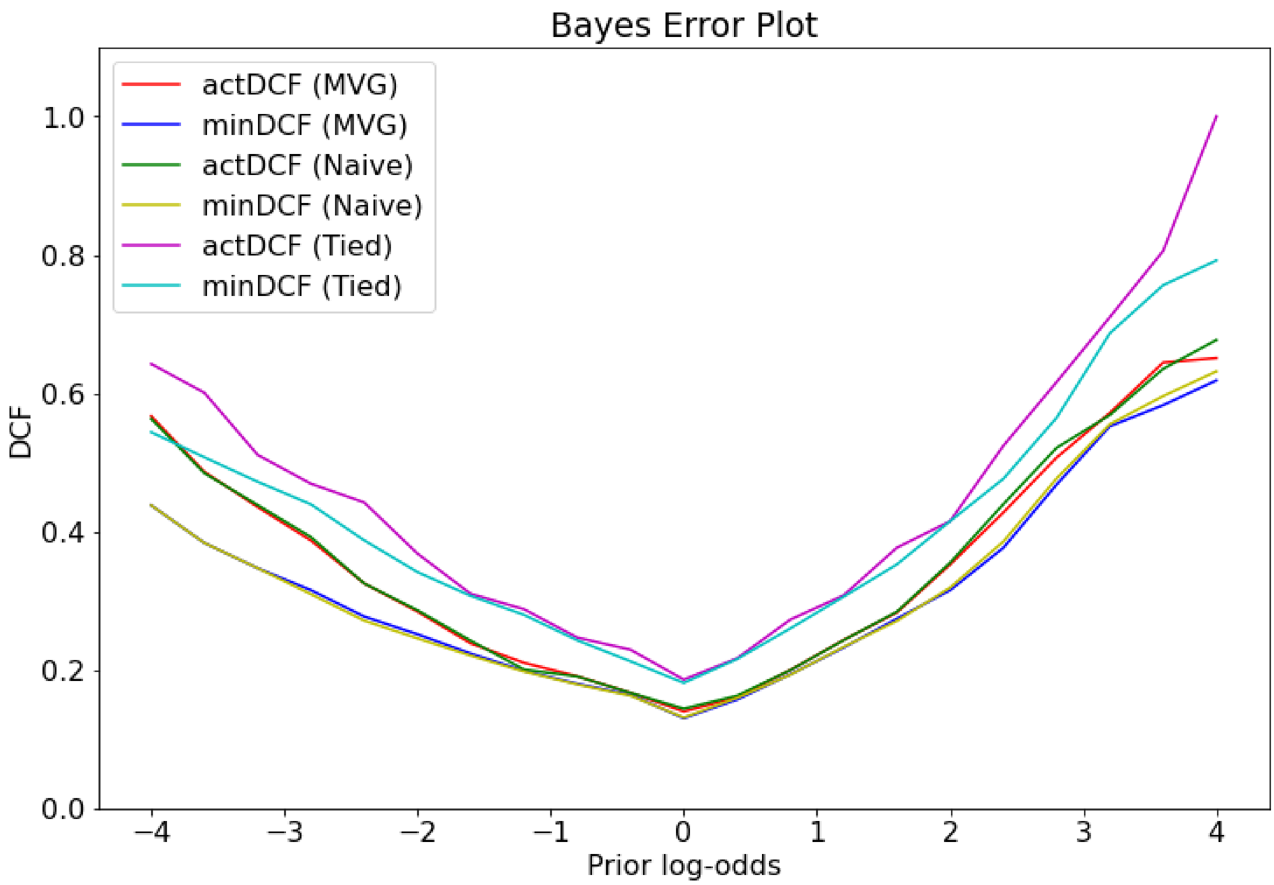
This is the best model that i found in terms of minDCF. This model seems well calibrated, as we can see by the low difference between actDCF and minDCF. Looking at the scores of all the models they all seem quite well calibrated (we don't see very big differences between DCF and minDCF).
Here are the other models with the same prior and costs.

Running experiment with Naive
Prior 0.5 - Cfn 1 - Cfp 1
[[925 77]
[67 931]]
DCF (non-normalized): 0.072
DCF (normalized): 0.144
MinDCF (normalized, fast): 0.131 (@ th = -1.791235e-01)

Running experiment with Tied
Prior 0.5 - Cfn 1 - Cfp 1
[[898 92]
[94 916]]
DCF (non-normalized): 0.093
DCF (normalized): 0.186
MinDCF (normalized, fast): 0.181 (@ th = -2.237531e-01)

Please note that we suppose a prior of 0.1 for all the other models, so for comparison we report the best MVG model with that prior.

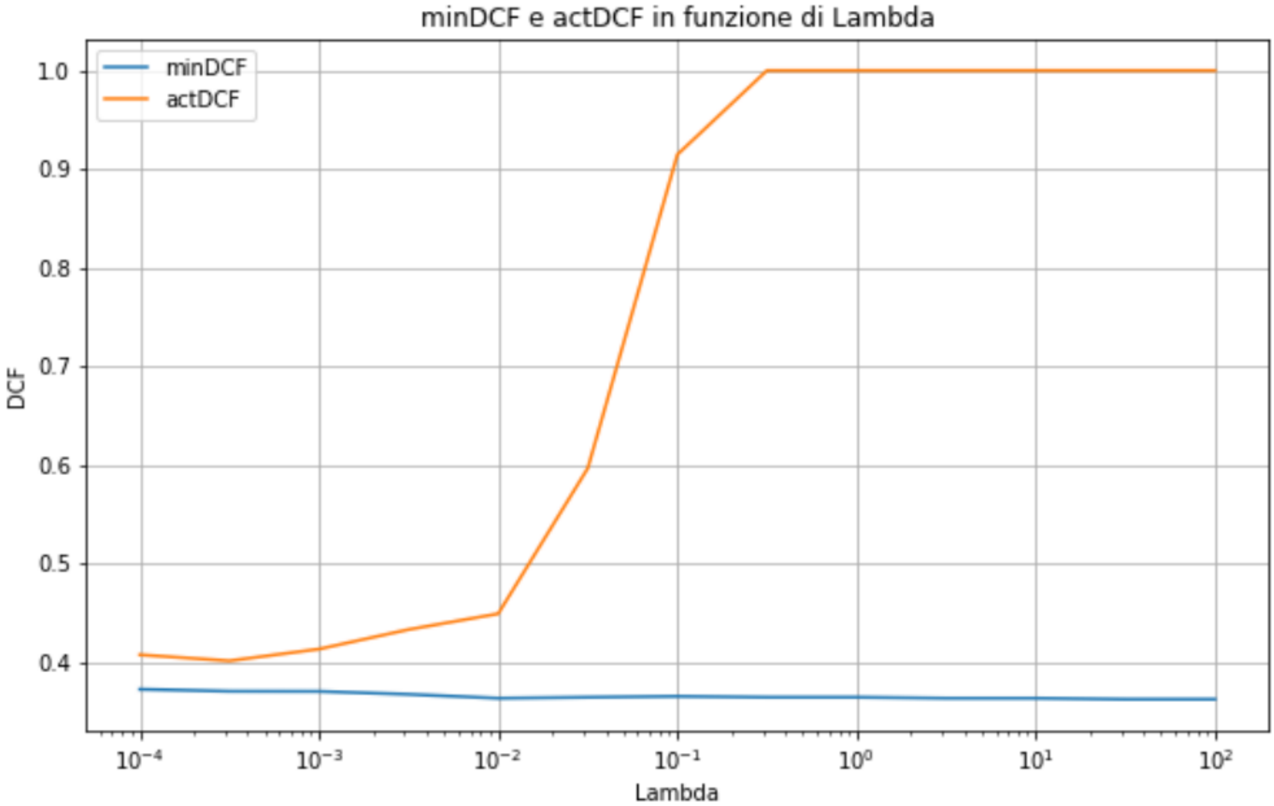
Prior 0.1 - Cfn 1 - Cfp 1
[[988 271]
[4 737]]
DCF (non-normalized): 0.031
DCF (normalized): 0.305
MinDCF (normalized, fast): 0.263 (@ th = 1.537195e+00)



Here we have the bayes plot of the best model found that we discussed above. The model seems well calibrated between -2 and 2, with some miscalibrations when approaching -4 and 4

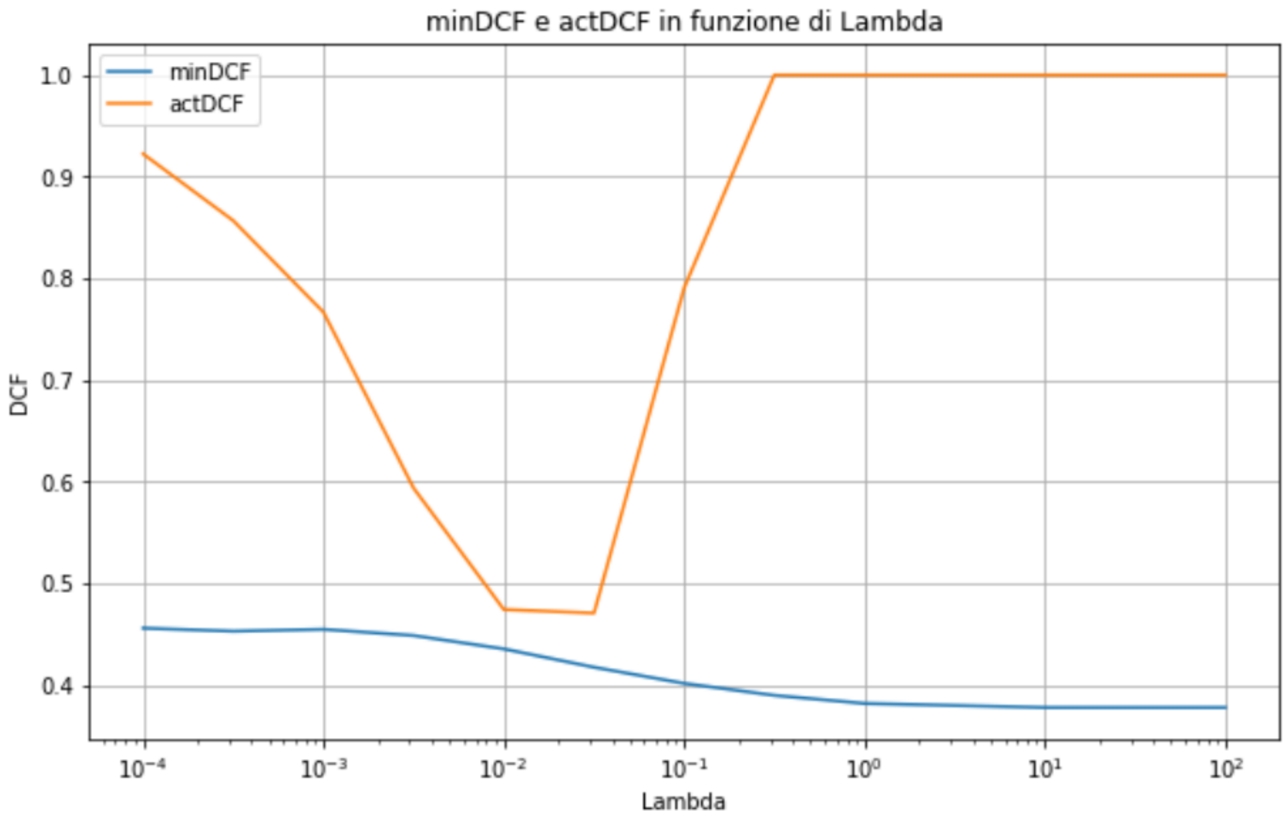
LAB8

Linear



We can notice a pretty big difference in actDCF with the increase of lambda. As lambda increases, we can notice excessive generalization of the model which causes a high actDCF value.

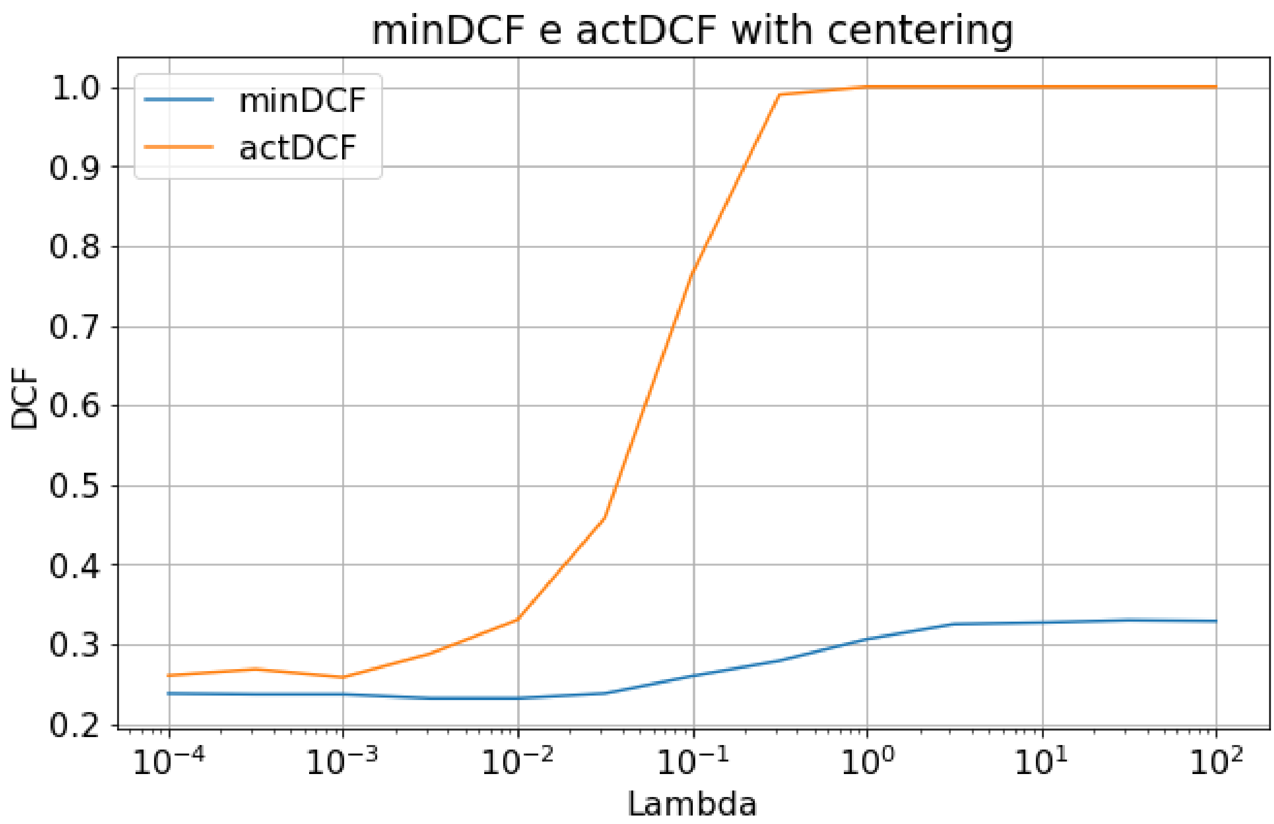
Limited Dataset



We can notice a pretty weird curve for the limited dataSet. Drops until 10^-2, then comes back up between 10^-2 and 10^-1. So for low values of lambda the model seem to struggle to obtain good separation for the data (this wasn't the case with the full dataset)

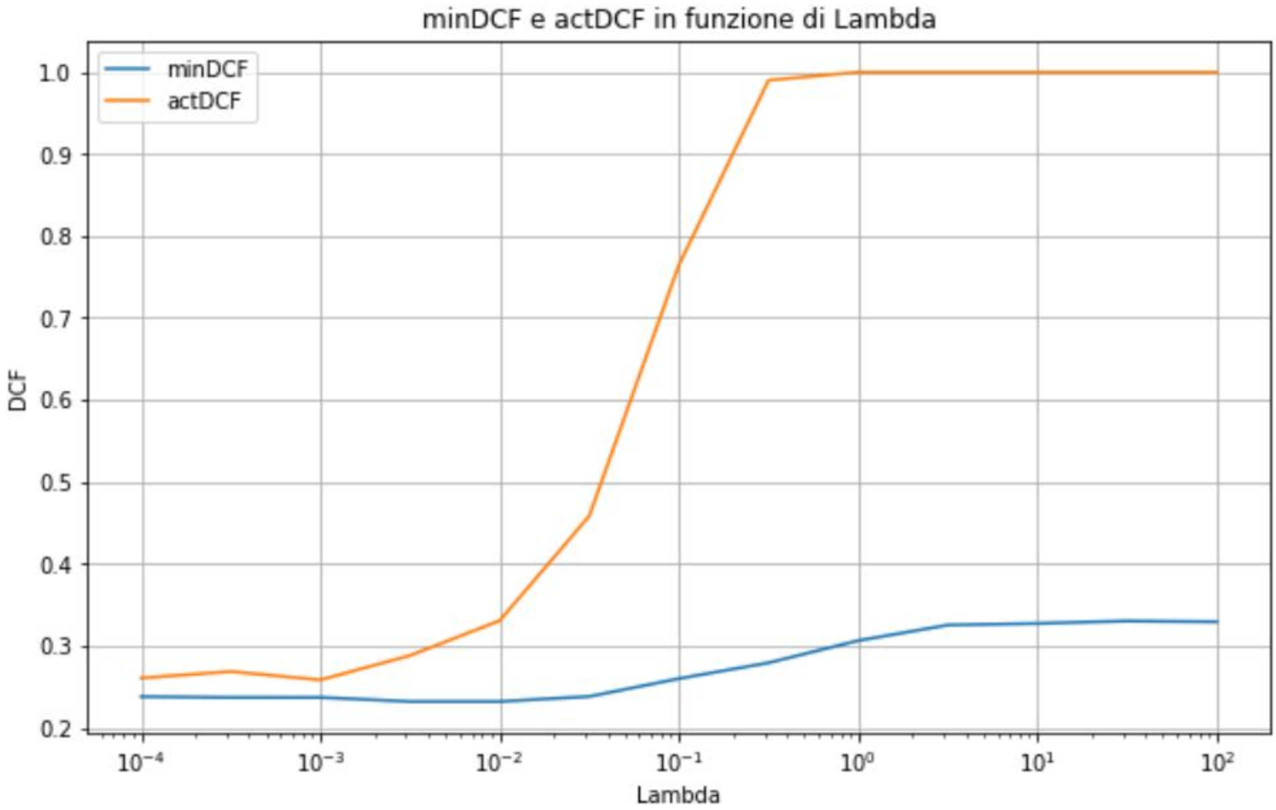
Between the non-weighted and weighted version we hardly notice any difference. Since the difference is not apparent, sticking with the non-weighted options seems best.

Centering



We can notice a slight change in the curve of actDCF and minDCF, but we can say that centering doesn't scramble as much the first result.

Quadratic model (best result)



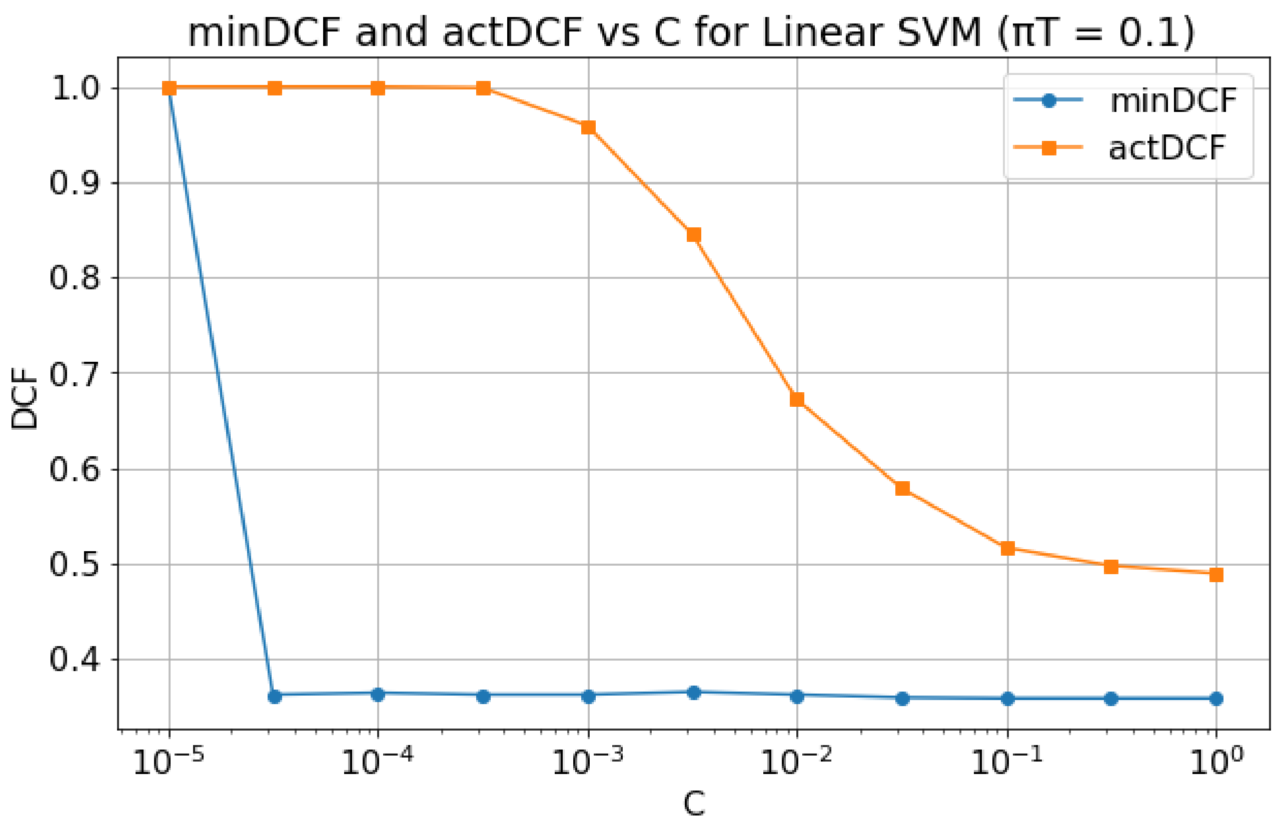
Best minDCF: 0.2320 with lambda = 3.1623e-03 achieved by the quadratic model.

In regard of the first plot, we can notice better overall minDCF and actDCF, with a model that's slightly more resistent to the increas of lambda (still, it ramps up around 10^-2, but still a better result than linear).

LAB9

Best linear SVM:

C: 0.1, minDCF: 0.3582, actDCF: 0.5162

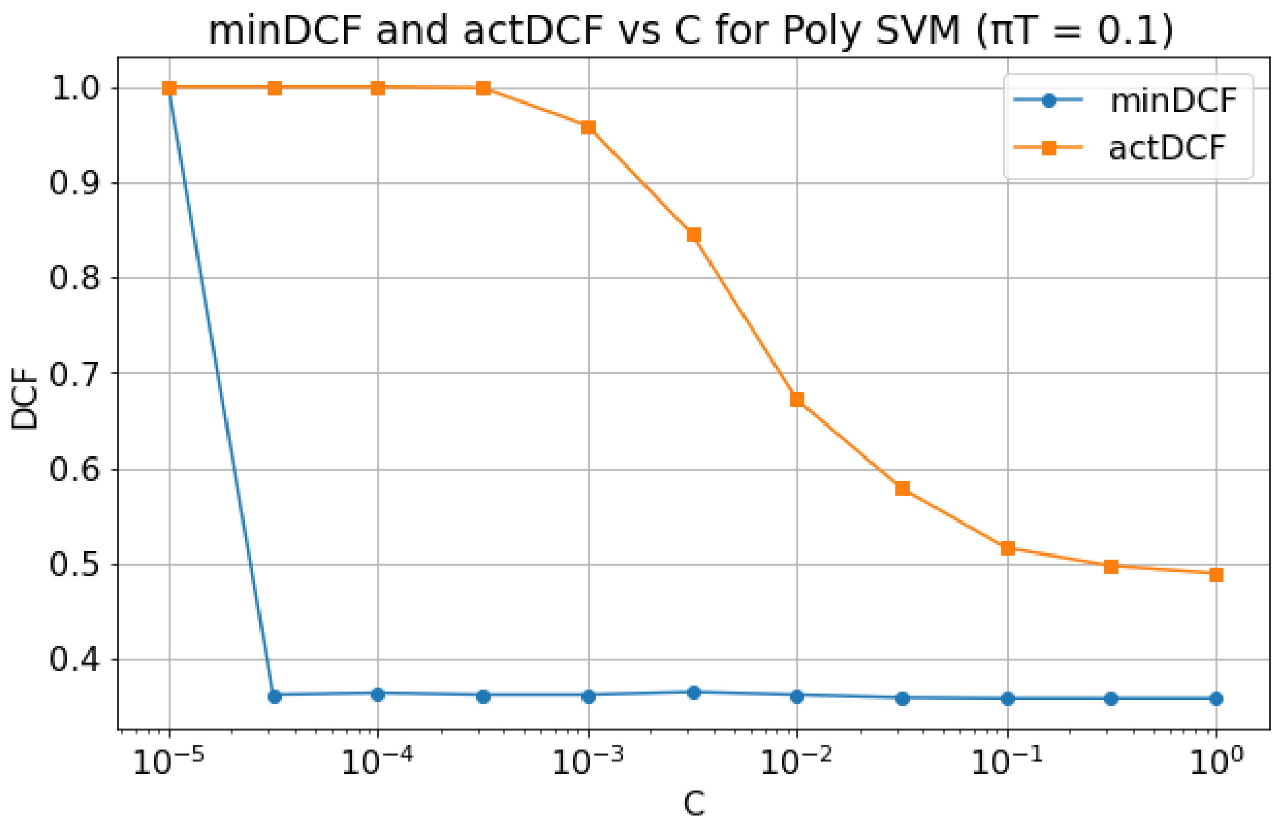


We can notice a spike in minDCF for the lowest value of C, so the most loose it can be (we allow a big number of points in the margin) , but then it stabilizes as C becomes stricter (and so the model doesn't do excessive generalization). We can notice a change in actDCF with the increase of C, slowly dropping from 1 to around 0.5.

It doesn't perform quite as good as the other models that we have considered (for example, Quad-LR has minDCF: 0.2320).

Best polinomial SVM (quadratic model):

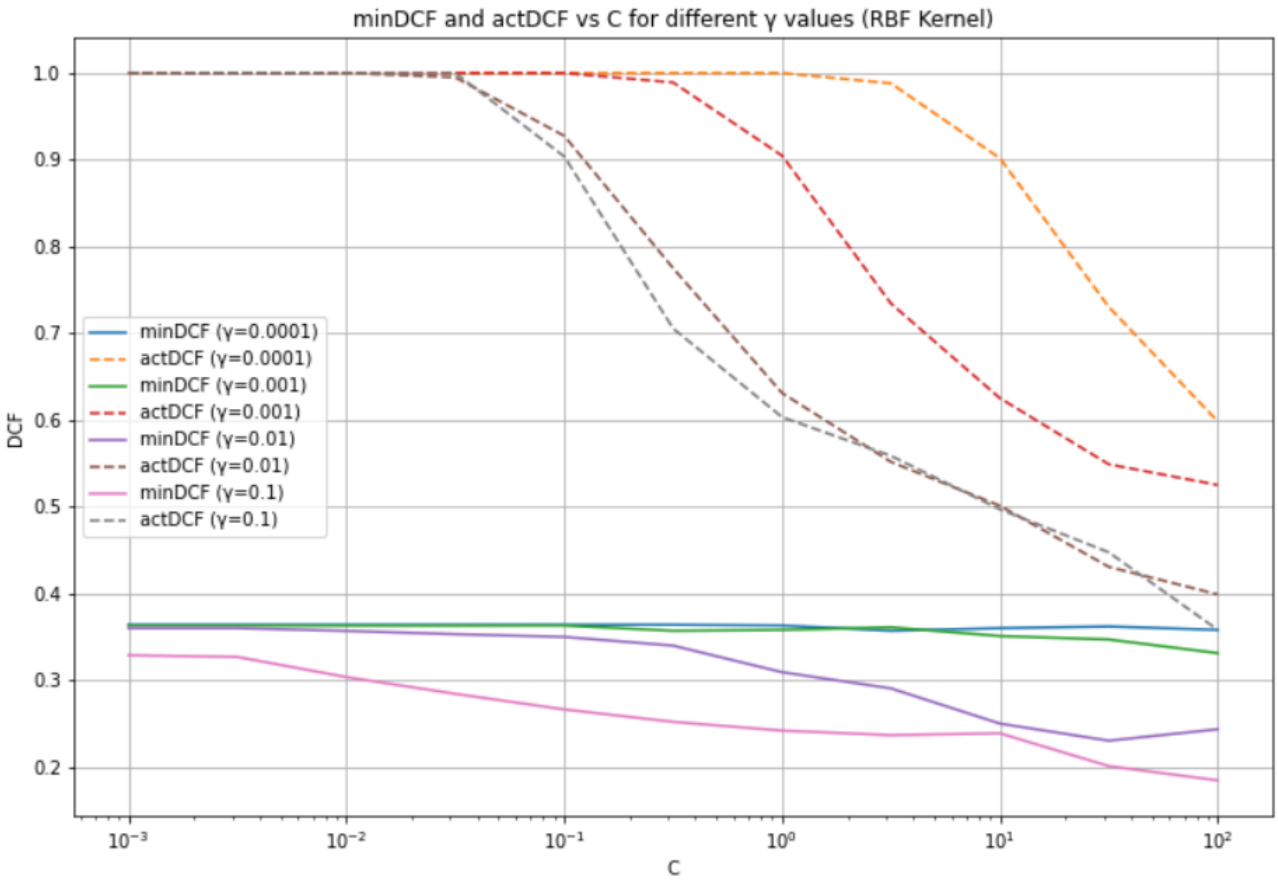
C: 0.01, minDCF: 0.1619, actDCF: 0.3056



This model presents a much better minDCF, which is now capable of competing with the other models that we've seen. ActDCF is still on the high side (probably because of the poor calibration)

Best RBF SVM:

gamma: 0.01, C: 31.622776601683793, minDCF: 0.1834, actDCF: 0.4137



Gamma and C do make a difference, as we can notice from the best model we've found. The scores doesn't appear well calibrated, with a big gap between each minDCF and actDCF. This model, although worse than poly kernel, is better than the linear SVM. The Quadratic-LR seems to have the upper hand on this model though.

LAB10

These are the best configurations that have been found.

Best configuration (full covariance): Class 0: 1 components, Class 1: 16 components

minDCF: 0.1495, actDCF: 0.2055

Best configuration (diagonal covariance): Class 0: 8 components, Class 1: 32 components

minDCF: 0.1312, actDCF: 0.1517

The best overall configuration for GMM is the one using the diagonal covariance. This result is surprising because the diagonal covariance has a reduced numbers of parameters to estimate in comparison of the full covariance, yet it achieves a better result.

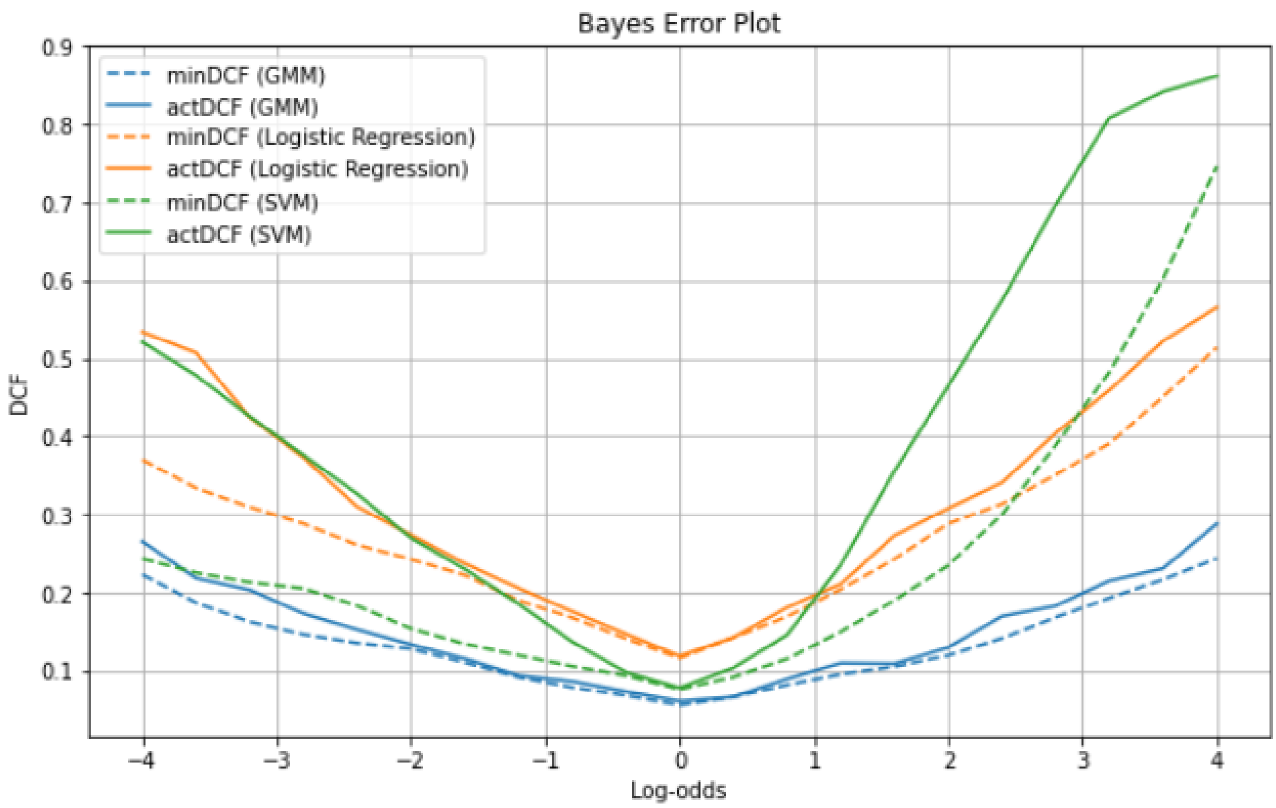
Model comparison for target application (prior = 0.1):

GMM: minDCF = 0.1312, actDCF = 0.1517

Quad-Logistic Regression: minDCF = 0.2507, actDCF = 0.2871

SVM: minDCF = 0.1668, actDCF = 0.3026

These are computed on the best configurations for each model. In terms of minDCF and actDCF, the most promising one is **GMM**



GMM seems to be by far the best calibrated, while SVM seems to be the worst one in terms of calibration. This happens because SVM uses a non-probabilistic approach,

LAB11

GMM: Validation set

minDCF(p=0.1), no cal.: 0.131

actDCF(p=0.1), no cal.: 0.152

minDCF(p=0.1), cal.   : 0.132

actDCF(p=0.1), cal.   : 0.152

SVM: Validation set

minDCF(p=0.1), no cal.: 0.167

actDCF(p=0.1), no cal.: 0.303

minDCF(p=0.1), cal.   : 0.172

actDCF(p=0.1), cal.   : 0.192

Validation set

minDCF(p=0.1), no cal.: 0.251

actDCF(p=0.1), no cal.: 0.287

minDCF(p=0.1), cal.   : 0.251

actDCF(p=0.1), cal.   : 0.266

calibration doesn't affect particularly the scores except for SVM, who has a drastic improvement in actDCF (matter of fact it was the one model than more than any other suffered from miscalibration)

This is very noticeable in the plot of calibrated SVM.

Fusion: Validation set

minDCF(p=0.1) : 0.125

actDCF(p=0.1) : 0.139

The fusion looks to be the best model so far. The scores are reasonably well calibrated, and the actDCF is better than any other model that we've seen. We will consider this as the delivered system, since it's the best performing in terms of minDCF and actDCF.

EVALUATION

Fusion: Evaluation set (delivered system)

minDCF(p=0.1) : 0.194

actDCF(p=0.1) : 0.197

GMM: Evaluation set

minDCF(p=0.1) : 0.183

actDCF(p=0.1), no cal.: 0.188

actDCF(p=0.1), cal. : 0.188

SVM:Evaluation set

minDCF(p=0.1) : 0.245

actDCF(p=0.1), no cal.: 0.285

actDCF(p=0.1), cal. : 0.256

LR:Evaluation set

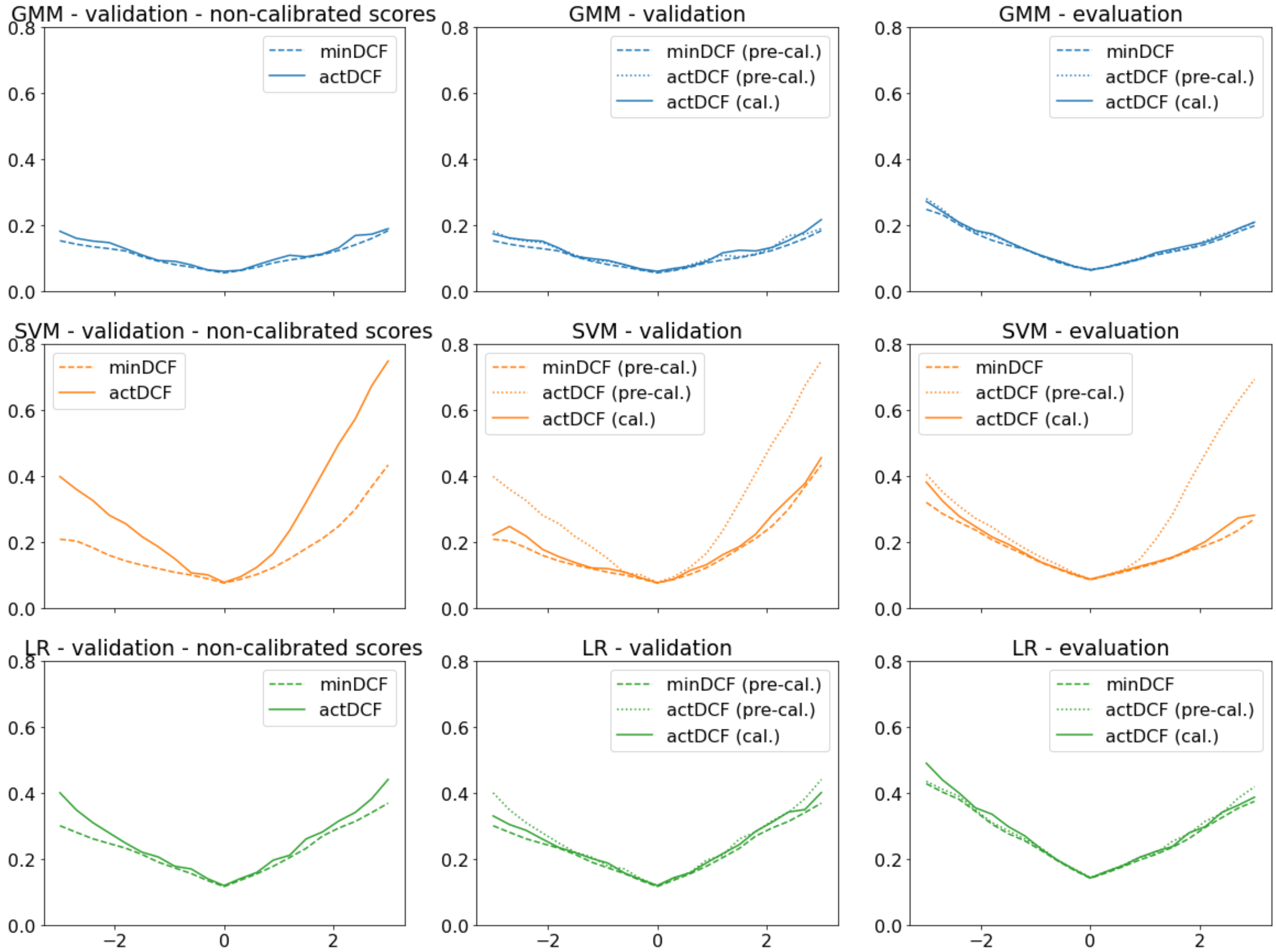
minDCF(p=0.1) : 0.356

actDCF(p=0.1), no cal.: 0.359

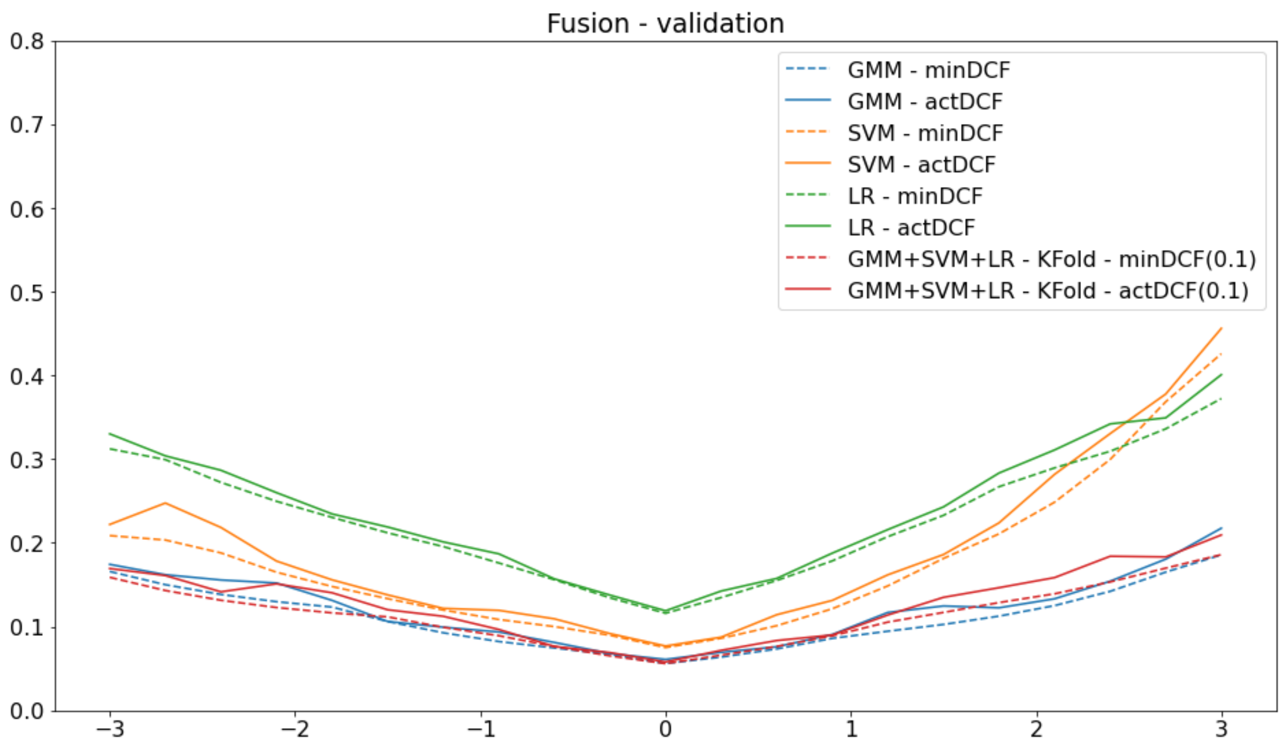
actDCF(p=0.1), cal. : 0.395

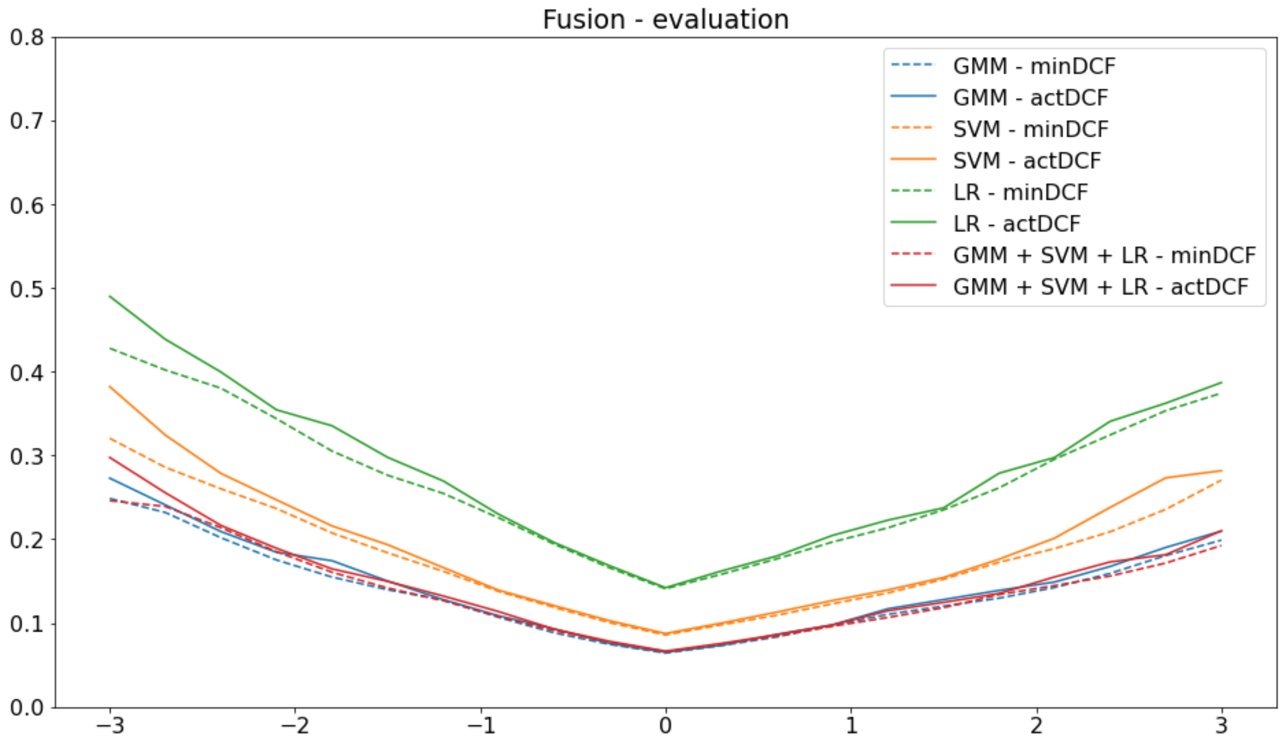
We can now see the perfomances of each of the three classifiers, pre/post calibration and also on the evaluation data.

K-fold

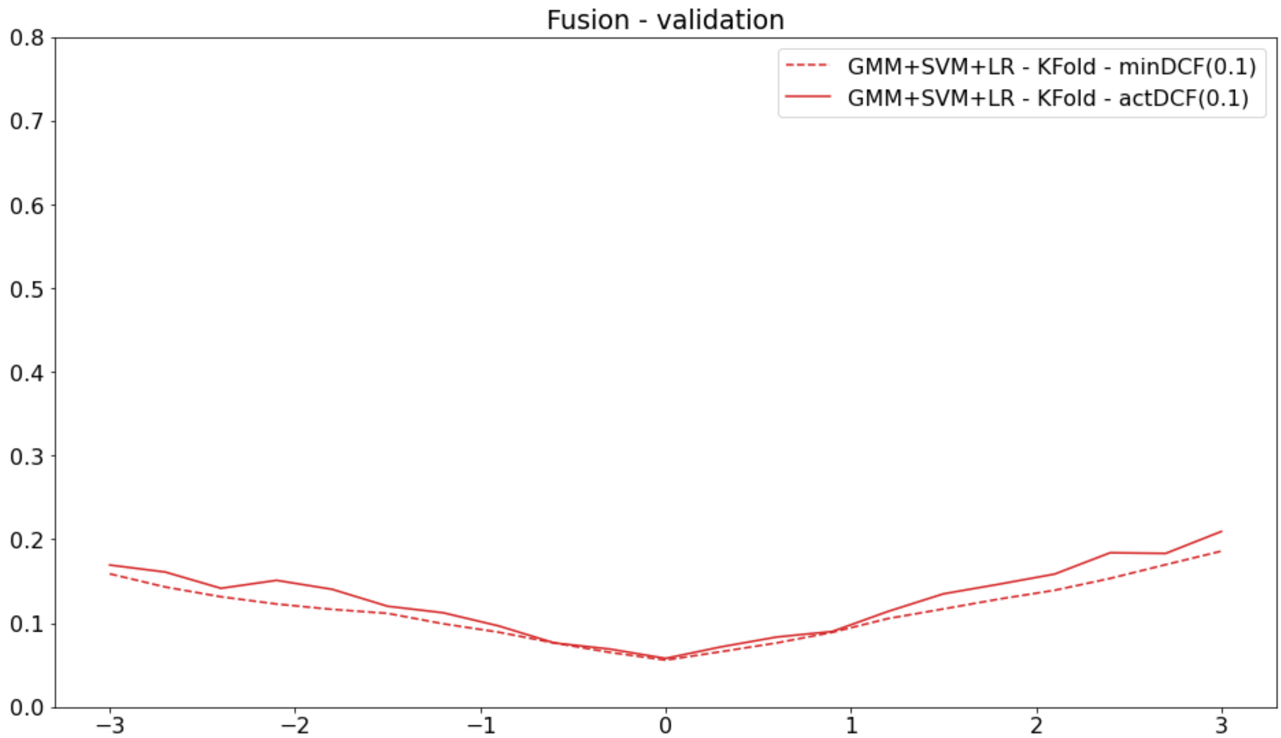


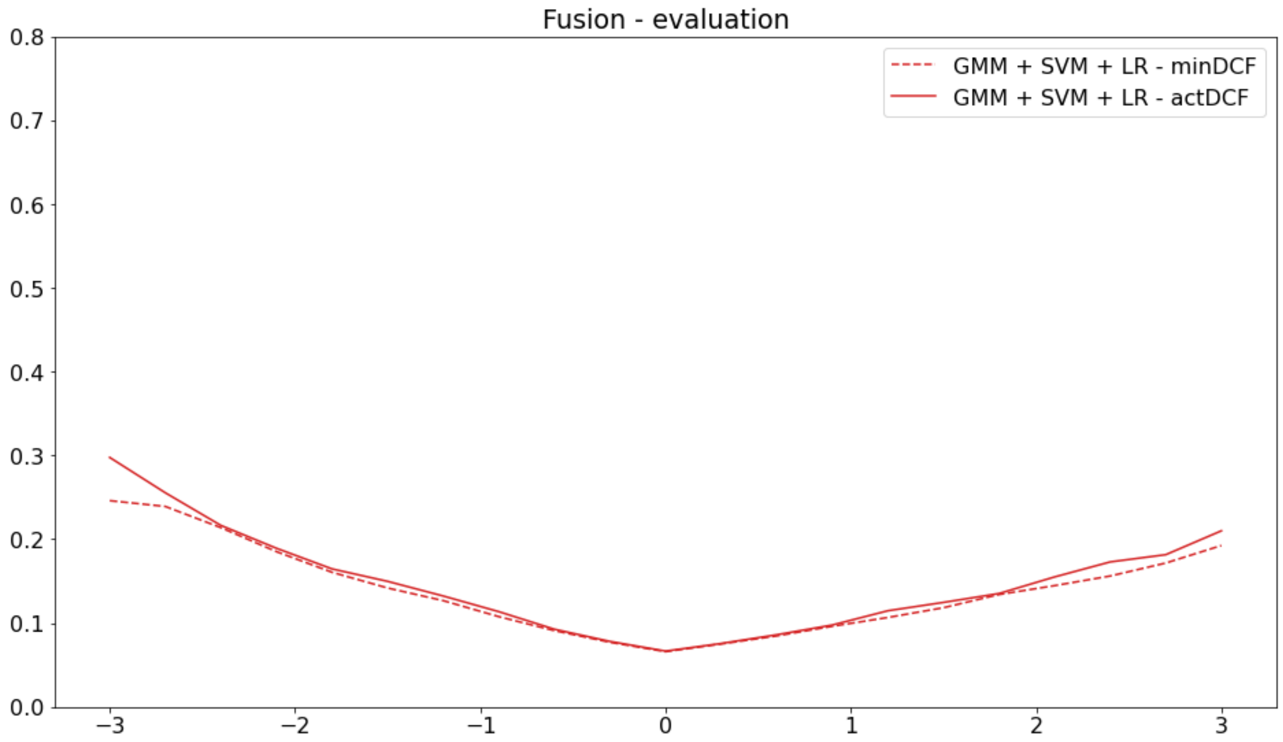
And now we can see a visualization with the fusion (delivered system)





And also we can see the fusion alone (delivered system)





We can notice a well calibrated model with the lowest minDCF so far.