Project Report

"Machine Learning and Pattern Recognition"

Politecnico di Torino

Academic Year 2023-2024

Marco Donnarumma s318922

1 Introduction

This project involves a binary classification task aimed at detecting fingerprint spoofing. The objective is to differentiate between genuine and counterfeit fingerprint images. The dataset includes labeled samples representing two classes: genuine fingerprints (True, label 1) and fake fingerprints (False, label 0). These samples are generated through a feature extractor that captures high-level characteristics of each fingerprint image. The resulting data is six-dimensional, capturing essential features to aid in the classification process.

2 Features Analysis - Laboratory 2

Feature	**Mean (True Class)**	**Mean (False Class)**
1	0.00054455	0.00287744
2	-0.00852437	0.01869316
3	0.66523785	-0.68094016
4	-0.66419535	0.6708362
5	-0.04172519	0.02795697
6	0.02393849	-0.0058274

Feature	**Variance (True Class)**	**Variance (False Class)**
1	1.43023345	0.56958105
2	0.57827792	1.42086571
3	0.5489026	0.54997702
4	0.55334275	0.53604266
5	1.3176792	0.6800736
6	1.28702609	0.70503844

Table 1: Mean and Variance for True and False Classes

The table presents the mean and variance for both 'True' and 'False' classes across six features. The means for the two classes are quite similar for most features, with differences most noticeable in features 3 and 4. For feature 3, the 'True' class has a positive mean (0.6652), while the 'False' class has a negative mean (-0.6809), indicating a significant separation between the classes for this feature. Similarly, feature 4 shows opposite signs for the means, with the 'True' class being negative (-0.6642) and the 'False' class being positive (0.6708), suggesting another point of differentiation.

Regarding variance, feature 1 has a significantly higher variance for the 'True' class (1.4302) compared to the 'False' class (0.5696), while for feature 2, the variance is higher for the 'False' class (1.4209) compared to the 'True' class (0.5783). This indicates that the spread of values for these features differs between the classes. Features 3 and 4 have similar variances across both classes, suggesting less differentiation. Features 5 and 6 show moderate differences in variance, with the 'True' class having higher variability in both cases. Overall, features 1, 2, 3, and 4 seem to provide some potential for distinguishing between the 'True' and 'False' classes, either through mean or variance differences.

2.1 Feature 1 and 2

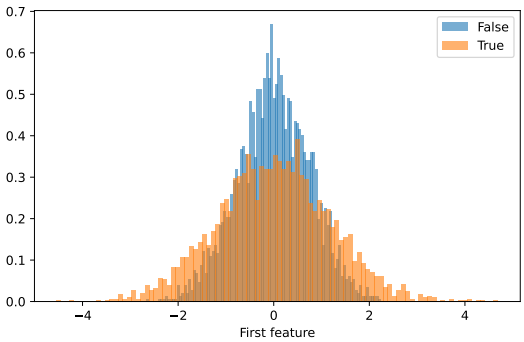

Figure 1: Histogram of Feature 1

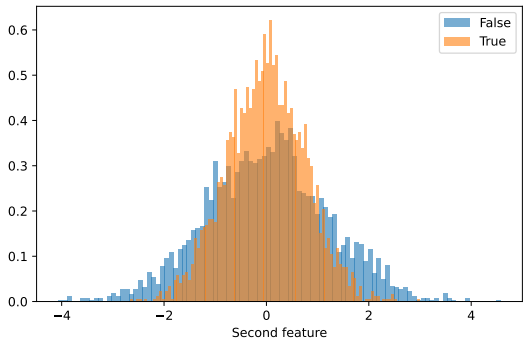

Figure 2: Histogram of Feature 2

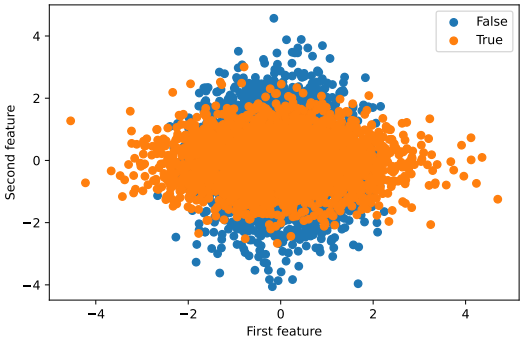

Figure 3: Scatter Plot of Feature 1 and 2

Based on the histograms of the first two features (Figures 1 and 2) and the scatter plot (Figure 3), we observe a noticeable overlap between the 'True' and 'False' classes. In Figure 1, which shows the histogram of the first feature, the 'True' class has a wider spread compared to the 'False' class, indicating that the 'True' class has a higher variance for this feature. Conversely, in Figure 2, which depicts the histogram of the second feature, the 'False' class exhibits a broader distribution, suggesting that it has a higher variance than the 'True' class for this feature. Both histograms are centered around zero, implying that the means of the two classes are quite similar for each feature. Additionally, each histogram presents a single peak, indicating unimodal distributions for both classes across both features. The scatter plot in Figure 3 further illustrates the overlap between the two classes, particularly around the origin, reinforcing that these two features alone may not provide sufficient discriminatory power to effectively separate the 'True' and 'False' classes.

2.2 Feature 3 and 4

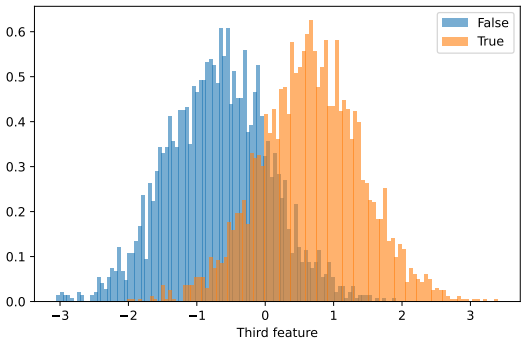

Figure 4: Histogram of Feature 3

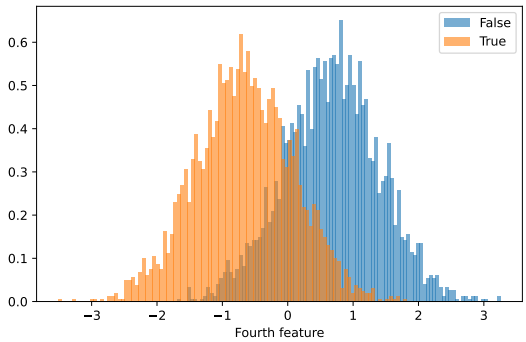

Figure 5: Histogram of Feature 4

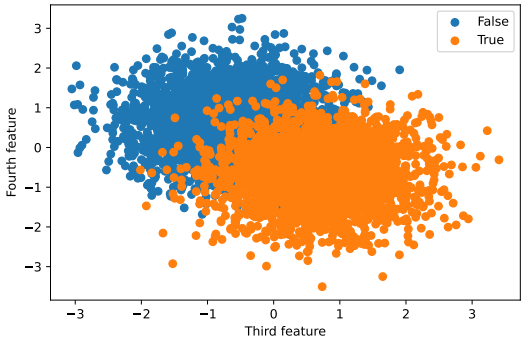

Figure 6: Scatter Plot of Feature 3 and 4

The histograms and scatter plot for the third and fourth features (Figures 4, 5, and 6) reveal a notable overlap between the 'True' and 'False' classes, although there are differences in their central tendencies. For feature 3, the 'False' class is more concentrated around negative values, while the 'True' class is centered around positive values, indicating distinct means for each class. In contrast, for feature 4, the 'True' class has a mean centered around negative values, while the 'False' class is centered around positive values, further emphasizing the difference in means. The variance appears to be relatively similar for both classes across each feature, as indicated by the similar spread in the histograms. Each histogram shows a single mode, confirming unimodal distributions. However, despite the separation in means, the significant overlap around the center suggests that these features alone may not provide strong discriminatory power between the classes.

2.3 Feature 5 and 6

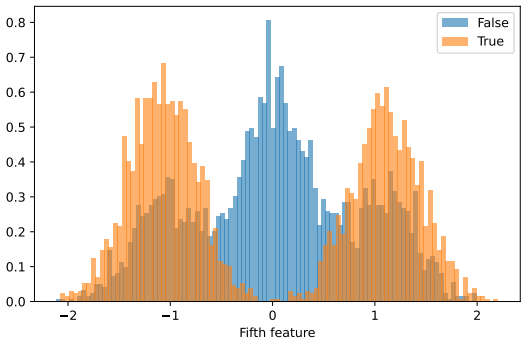

Figure 7: Histogram of Feature 5

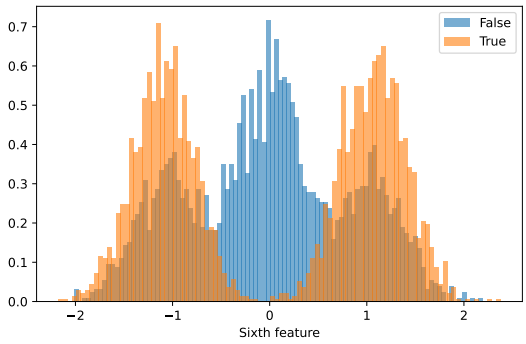

Figure 8: Histogram of Feature 6

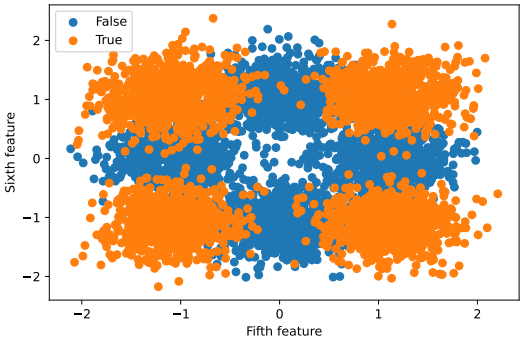

Figure 9: Scatter Plot of Feature 5 and 6

Based on the histograms and scatter plot for the fifth and sixth features (Figures 7, 8, and 9), we observe that there is a considerable overlap between the 'True' and 'False' classes. In both histograms, the classes overlap around the center, particularly near zero, indicating that neither feature alone provides a clear separation between the classes. The histograms for features 5 and 6 show multiple modes, suggesting a multimodal distribution, which may indicate underlying subgroups within each class. The scatter plot (Figure 9) further supports this observation, revealing multiple clusters for each class; there appear to be four distinct clusters, suggesting that both features combined capture more complex patterns or groupings within the data that are not evident from the histograms alone.

3 Dimensionality Reduction - Laboratory 3

3.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) was performed on the dataset to examine its impact on the distribution of features. The initial PCA components account for most of the variance, with the first component demonstrating a clearer separation between classes than the subsequent components.

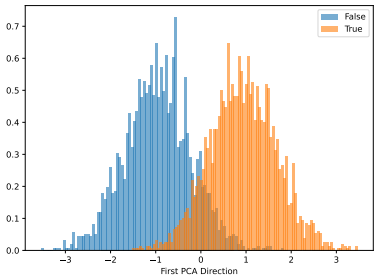

Figure 10: First PCA direction

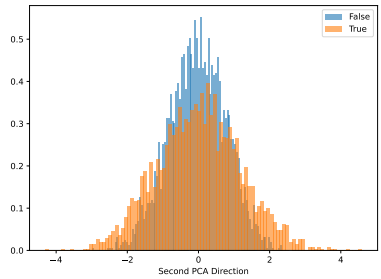

Figure 11: Second PCA direction

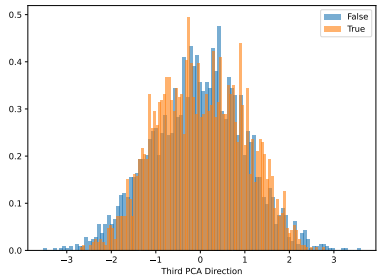

Figure 12: Third PCA direction

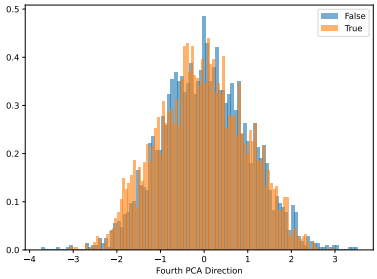

Figure 13: Fourth PCA direction

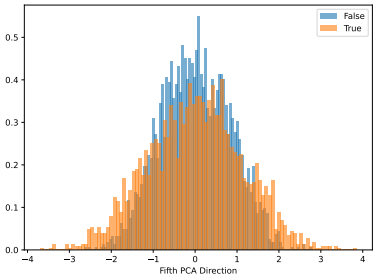

Figure 14: Fifth PCA direction

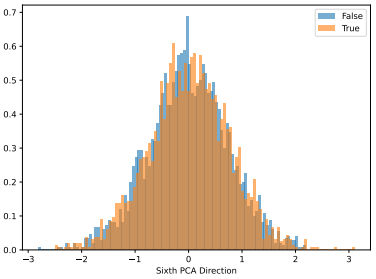

Figure 15: Sixth PCA direction

The histograms for the six PCA directions (Figures 10-15) show how the projected features vary along each principal component, with the first component capturing the most variance. In the first PCA direction (Figure 10), there is a noticeable separation between the 'True' and 'False' classes, indicating that this component captures a significant amount of the variance that distinguishes the classes. As we move to the subsequent PCA directions (Figures 11-15), the overlap between the two classes increases, and the separation becomes less pronounced, reflecting that these components capture less variance relevant to class distinction. Overall, the first PCA component provides the most distinct separation between classes, while the later components contribute less to class discrimination, and no distinct clusters are observed within each class.

3.2 Linear Discriminant Analysis (LDA)

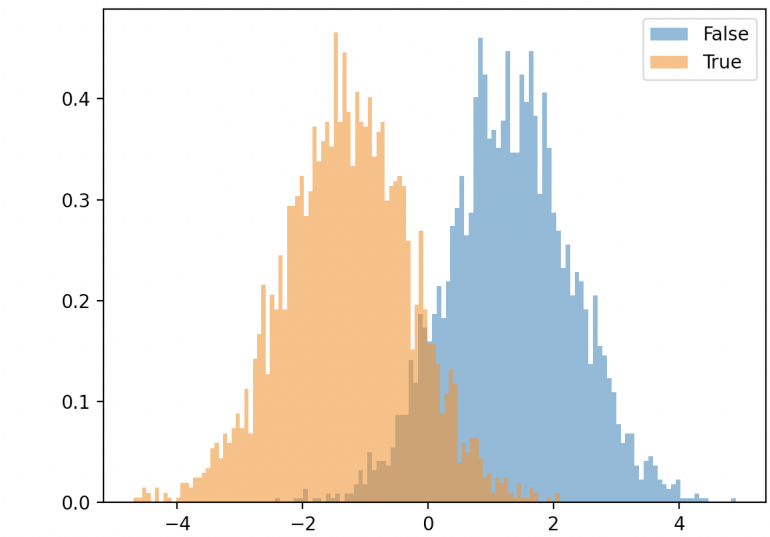

Figure 16: LDA Projected Samples

Based on the histogram of the LDA-projected samples (Figure 16), we observe that the 'True' and 'False' classes are separated along a single LDA dimension, though there is still some overlap between them. Compared to the histograms of the six PCA directions, the LDA projection provides a clear separation between the two classes, similar to the first PCA direction, which also showed a good degree of separation. However, the key difference is that LDA explicitly finds the direction that maximizes class separation, whereas PCA's first component captures the direction of maximum variance without necessarily focusing on class discrimination. Thus, while both LDA and the first PCA direction achieve a similar separation, LDA does so with the specific aim of minimizing class overlap.

3.3 Classification with LDA

The evaluation of LDA as a classifier was conducted by splitting the data into training and validation sets. Initially, the decision threshold was set at the mean of the class projections, which resulted in 1,814 correct predictions out of 2,000 validation samples, corresponding to an error rate of 9.30%. Adjusting the threshold slightly improved the results: increasing the threshold by 0.01 lowered the error rate to 9.25%, while decreasing it by 0.01 reduced the error to 9.20%. These findings indicate that LDA effectively separates the classes, but fine-tuning the decision threshold is essential for optimal performance, as minor adjustments can enhance accuracy. This suggests that while LDA performs well, some overlap between class distributions remains near the decision boundary.

3.4 Classification using LDA with PCA Pre-processing

Based on the results in the table, we observe that varying the number of PCA dimensions m does not significantly impact the performance of the LDA classifier on the validation set. The error rates remain very close, ranging from 9.25% to 9.35% across different values of m. The lowest error rate (9.25%) is observed for m=2,3, and 4, but this is only a slight improvement compared to the other values.

From these results, it appears that pre-processing the features with PCA does not provide a substantial benefit when combined with the LDA classifier for this task. The LDA classifier's performance remains relatively stable regardless of the number of PCA components retained, suggesting that the original feature set (without dimensionality reduction) is already adequate for separating the classes. Thus, applying PCA in this case does not seem to significantly improve accuracy.

m	**Error Rate**
5	9.3%
4	9.25%
3	9.25%
2	9.25%
1	9.35%

Table 2: Results PCA + LDA

4 Multivariate Gaussian Density - Laboratory 4

Authenticated Fingerprints Density Plot

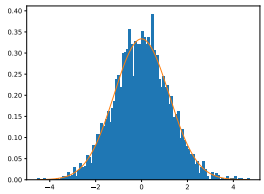

Figure 17: Feature 1

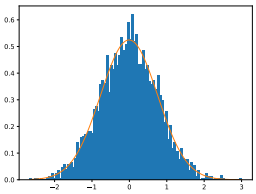

Figure 18: Feature 2

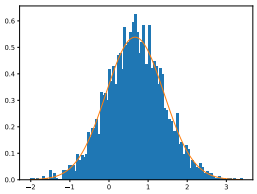

Figure 19: Feature 3

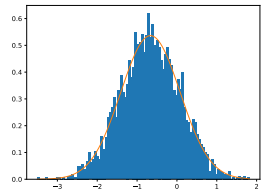

Figure 20: Feature 4

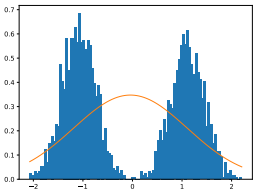

Figure 21: Feature 5

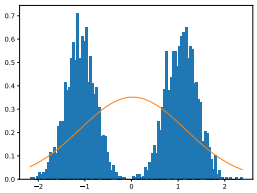

Figure 22: Feature 6

Spoofed Fingerprints Density Plot

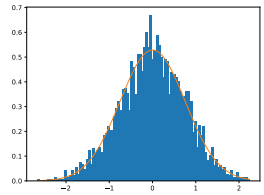

Figure 23: Feature 1

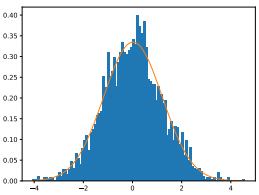

Figure 24: Feature 2

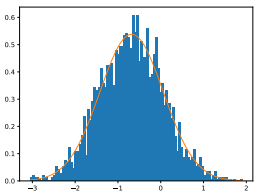

Figure 25: Feature 3

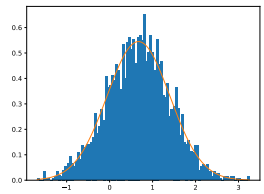

Figure 26: Feature 4

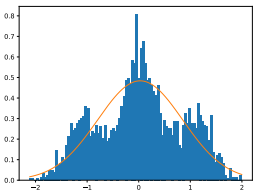

Figure 27: Feature 5

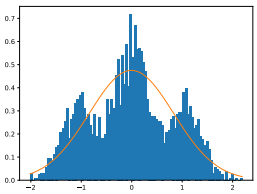

Figure 28: Feature 6

The plots show the density distributions of features from fingerprint data, divided into "Authenticated Fingerprints" and "Spoofed Fingerprints." For both categories, Features 1 to 4 exhibit a unimodal, bell-shaped distribution that fits well with the Gaussian density curve, indicating that a Gaussian model is appropriate for these features. However, Features 5 and 6 present different challenges: while the "Authenticated Fingerprints" display a bimodal distribution with two peaks, the "Spoofed Fingerprints" show a more complex, trimodal distribution with three distinct peaks. This complexity makes the Gaussian model an inadequate fit for Features 5 and 6, particularly in the "Spoofed Fingerprints," where the Gaussian assumption of a single peak fails to capture the data's multi-peaked nature.

5 Gaussian Models for Binary Classification - Laboratory 5

Model	Error Rate
LDA	**9.3%**
MVG	**7.0%**
Naive Bayes Gaussian	**7.2%**
Tied Gaussian	**9.3%**

Table 3: Error rates for different classifiers.

5.1 Models comparison

The MVG (Multivariate Gaussian) classifier achieves the lowest error rate of 7.0%, indicating its superior ability to capture complex feature dependencies due to its use of a full covariance matrix. The Naive Bayes Gaussian classifier follows with a slightly higher error rate of 7.2%, suggesting that while it simplifies the model by assuming feature independence, it still performs relatively well. Both the LDA (Linear Discriminant Analysis) and Tied Gaussian classifiers have the highest error rates at 9.3%, indicating that these models struggle with non-linear class boundaries and that the assumption of a common covariance matrix across classes is not suitable for this dataset.

5.2 Pearson Correlation Covariance Analysis

1.0000	5.5352e-05	3.2698e-02	3.3747e-02	1.9797e-02	-2.0974e-02	-3.3747e-02
5.5352e-05	1.0000	-1.7837e-02	-1.7909e-02	-2.6356e-02	2.2983e-02	2.2257e-02
3.2698e-02	-1.7837e-02	1.0000	-3.3314e-03	-1.1022e-02	2.7116e-02	-3.3314e-03
3.3747e-02	-1.7909e-02	-3.3314e-03	1.0000	8.5532e-03	2.2257e-02	8.5532e-03
1.9797e-02	-2.6356e-02	-1.1022e-02	8.5532e-03	1.0000	2.2919e-02	2.2919e-02
-2.0974e-02	2.2983e-02	2.7116e-02	2.2257e-02	2.2919e-02	1.0000	1.0000

Table 4: Correlation Matrix for Class False

1.0000	-1.6446e-02	6.1994e-03	1.7332e-02	1.3986e-02	-1.2859e-04	-2.0974e-02
-1.6446e-02	1.0000	-2.0195e-02	-1.6145e-02	-1.7010e-02	1.9248e-02	-1.6446e-02
6.1994e-03	-2.0195e-02	1.0000	4.8907e-02	-4.3582e-02	-1.3379e-02	-4.3582e-02
1.7332e-02	-1.6145e-02	4.8907e-02	1.0000	4.0605e-02	4.0605e-02	-3.3314e-03
1.3986e-02	-1.7010e-02	-4.3582e-02	4.0605e-02	1.0000	1.2811e-02	1.2811e-02
-1.2859e-04	1.9248e-02	-1.7123e-02	4.0605e-02	1.2810e-02	1.0000	1.0000

Table 5: Correlation Matrix for Class True

The correlation matrices for the "False" and "True" classes reveal that most feature pairs exhibit very low correlation values, close to zero, indicating minimal linear relationships between features within each class. The diagonal elements are 1, as expected, while the off-diagonal elements mostly show weak positive or negative correlations, suggesting that the features are largely independent or only weakly correlated. This observation aligns with the assumptions of simpler models like the Naive Bayes Gaussian classifier, which assumes feature independence. However, even these small correlations contribute to the superior performance of the MVG classifier, which accounts for full covariance between features, unlike models such as LDA and Tied Gaussian that assume shared covariance or linear separability, leading to their relatively higher error rates.

5.3 Analysis using only the first 4 features

Model	Error Rate (All features)	Error Rate (First 4 features)
MVG	**7.0%**	**8.0%**
Naive Bayes Gaussian	**7.2%**	**7.6%**
Tied Gaussian	**9.3%**	**9.5%**

Table 6: Error rates for different classifiers.

Based on the table, when the classification is repeated using only the first four features (i.e., discarding the last two features), the error rates for all three models slightly increase. The MVG classifier's error rate increases from 7.0% to 8.0%, the Naive Bayes Gaussian classifier's error rate increases from 7.2% to 7.6%, and the Tied Gaussian classifier's error rate rises from 9.3% to 9.5%. This suggests that, despite the poor fit of the Gaussian assumptions for the last two features, these features still contain some useful information

that contributes to improving classification accuracy. Discarding the last two features results in a decline in performance across all models, indicating that the Gaussian models are able to extract valuable information from these features, even if the assumptions about their distribution are not entirely accurate.

5.4 Analysis pair of features

5.4.1 Feature 1 and 2

Model	Error Rate
MVG	**36.5%**
Naive Bayes Gaussian	**36.3%**
Tied Gaussian	**49.5%**

Table 7: Error rates for feature 1 and 2

5.4.2 Feature 3 and 4

Model	Error Rate
MVG	**9.4%**
Naive Bayes Gaussian	**9.4%**
Tied Gaussian	**9.4%**

Table 8: Error rates for feature 3 and 4

As noted in the initial Feature Analysis, features 3 and 4 show a clearer separation between the classes, with identical variances but different means. In contrast, features 1 and 2 have similar means across classes but differ in their variances. From the previous results, it is clear that features 3 and 4 provide more valuable information for distinguishing between classes. Interestingly, the Tied model performs similarly to other models when using these two features, aligning with its assumption of equal variances across classes.

On the other hand, features 1 and 2 demonstrate poorer classification performance across all models and violate the Tied model's equal variance assumption, leading to suboptimal results for this model when using these features.

This analysis highlights the distinctions between features 1-2 and 3-4 and emphasizes the impact of the variance assumption on the Tied model's performance.

5.5 PCA Pre-processing

Model	m = 1	m = 2	m = 3	m = 4	m = 5	m = 6
MVG	**9.2%**	**8.8%**	**8.8%**	**8.1%**	**7.1%**	**7.0%**
Naive Bayes Gaussian	**9.2%**	**8.8%**	**9.0%**	**8.8%**	**8.8%**	**8.9%**
Tied Gaussian	**9.3%**	**9.2%**	**9.2%**	**9.2%**	**9.3%**	**9.3%**

Table 9: Error rates for different values of m.

The table presents the error rates for three classifiers — MVG, Naive Bayes Gaussian, and Tied Gaussian — across different values of m, representing the number of PCA components. As m increases, the MVG classifier's error rate steadily decreases, reaching the lowest value of 7.0% at m=6, indicating that it effectively leverages the additional information provided by more PCA components. The Naive Bayes Gaussian classifier shows slight variations in error rates, generally remaining stable but with a slight increase when m=6. The Tied Gaussian classifier, however, exhibits relatively consistent error rates across all values of m, suggesting that it is less influenced by the number of PCA components. Overall, the table demonstrates that while all classifiers benefit from PCA, the MVG classifier shows the most significant improvement as the number of components increases.

6 Model Performance Across Applications - Laboratory 7

This section analyzes the performance of the MVG, Tied Gaussian, and Naive Gaussian models across three different applications, each characterized by effective priors of $\tilde{\pi} = 0.1$, $\tilde{\pi} = 0.5$, and $\tilde{\pi} = 0.9$. For each application, the optimal Bayes decisions on the validation set were computed, both with and without PCA as a pre-processing step, while varying the number of components (m).

Model	PCA (m)	Actual DCF	Minimum DCF	Calibration Loss
MVG	None	0.3051	0.2629	13.82%
	5	0.3041	0.2738	9.96%
	4	0.3529	0.3012	14.65%
	3	0.3879	0.3563	8.16%
	2	0.3879	0.3526	9.09%
	1	0.3969	0.3686	7.12%
Tied	None	0.4061	0.3628	10.65%
	5	0.4050	0.3648	9.93%
	4	0.4030	0.3609	10.40%
	3	0.4081	0.3680	9.82%
	2	0.3959	0.3630	8.30%
	1	0.4020	0.3686	8.30%
Naive	None	0.3021	0.2569	14.94%
	5	0.3930	0.3544	9.82%
	4	0.3970	0.3614	8.97%
	3	0.3950	0.3645	7.71%
	2	0.3869	0.3561	7.96%
	1	0.3969	0.3686	7.12%

Table 10: Effective prior $\tilde{\pi} = 0.1$ with Calibration Loss.

Model	PCA (m)	Actual DCF	Minimum DCF	Calibration Loss
MVG	None	0.1399	0.1302	6.93%
	5	0.1419	0.1331	6.19%
	4	0.1609	0.1537	4.48%
	3	0.1759	0.1734	1.42%
	2	0.1760	0.1731	1.65%
	1	0.1850	0.1769	4.38%
Tied	None	0.1860	0.1812	2.58%
	5	0.1860	0.1812	2.58%
	4	0.1850	0.1821	1.57%
	3	0.1850	0.1830	1.08%
	2	0.1850	0.1789	3.30%
	1	0.1870	0.1769	5.38%
Naive	None	0.1439	0.1311	8.89%
	5	0.1750	0.1737	0.74%
	4	0.1770	0.1717	2.99%
	3	0.1799	0.1746	2.95%
	2	0.1770	0.1710	3.39%
	1	0.1850	0.1769	4.38%

Table 11: Effective prior $\tilde{\pi} = 0.5$ with Calibration Loss.

Model	PCA (m)	Actual DCF	Minimum DCF	Calibration Loss
MVG	None	0.4001	0.3423	14.44%
	5	0.3980	0.3512	11.75%
	4	0.4598	0.4150	9.73%
	3	0.4680	0.4392	6.16%
	2	0.4434	0.4384	1.13%
	1	0.4775	0.4341	9.09%
Tied	None	0.4626	0.4421	4.42%
	5	0.4626	0.4451	3.79%
	4	0.4615	0.4441	3.76%
	3	0.4565	0.4342	4.88%
	2	0.4785	0.4352	9.05%
	1	0.4805	0.4341	9.66%
Naive	None	0.3893	0.3510	9.84%
	5	0.4660	0.4340	6.87%
	4	0.4630	0.4313	6.84%
	3	0.4591	0.4343	5.40%
	2	0.4424	0.4323	2.28%
	1	0.4775	0.4341	9.09%

Table 12: Effective prior $\tilde{\pi} = 0.9$ with Calibration Loss.

The MVG model consistently achieved the lowest Minimum DCF across all applications, both with and without PCA, demonstrating its superior performance. However, its effectiveness declined with lower m values (below 5) when PCA was applied, suggesting that PCA might remove important discriminative information necessary for the MVG model's full covariance matrices. The Tied Gaussian model underperformed compared to the MVG model across all applications, likely due to its assumption of shared covariance matrices across classes, which does not fit this dataset. Although the Tied model's performance was stable across different m values, it remained less effective overall.

The Naive Gaussian model showed mixed results, performing nearly as well as the MVG model for extreme priors ($\tilde{\pi} = 0.1$ or $\tilde{\pi} = 0.9$) but declining with PCA application, albeit with better calibration than the Tied model. This suggests that the Naive model's independence assumption can be beneficial when not heavily violated by the data.

In conclusion, the MVG model with m=5 PCA dimensions is the best-performing option across all applications that use PCA. While PCA offers some advantages, it should be used carefully with complex models like the MVG. The Naive Gaussian model provides a simpler, competitive alternative for extreme priors, while the Tied Gaussian model remains limited due to its restrictive assumptions.

6.1 Bayes Error Analysis

This section presents the Bayes error plots for the MVG, Naive Bayes, and Tied Gaussian classifiers, evaluated under the primary application setting with an effective prior of $\tilde{\pi} = 0.1$. The plots cover a range of prior log-odds from -4 to $+4$, allowing us to compare the Minimum and Actual DCF for each model and assess their calibration and consistency across different applications. The PCA setup that gave the best results was for m = 5, we therefore analyze this scenario.

An analysis of the Bayes error plots shows that the MVG model consistently achieves the lowest Minimum DCF across the entire range of prior log-odds, demonstrating its robust and stable performance across various application scenarios. Additionally, the

Actual DCF remains close to the Minimum DCF, indicating that the MVG model is well-calibrated and reliable for the intended applications.

The Naive Bayes model also performs well, with a Minimum DCF only slightly higher than that of the MVG model. However, its Actual DCF deviates more from the Minimum DCF in certain regions, suggesting calibration issues under specific prior log-odds settings. In contrast, the Tied Gaussian model exhibits the highest Minimum DCF in all cases, with a larger gap between the Actual and Minimum DCF, indicating poor calibration and weaker overall performance, especially at the extremes of the prior log-odds range. This makes the Tied Gaussian model less suitable for applications requiring accurate probability estimation and adaptability to varying priors.

In summary, the MVG model is the most reliable and well-calibrated choice, with the Naive Bayes model as a close alternative. The Tied Gaussian model, while potentially useful in some scenarios, is less effective for applications demanding high accuracy and robust calibration.

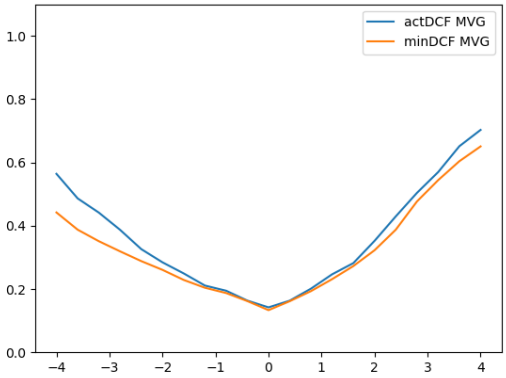

Figure 29: MVG - Bayes Error Plot

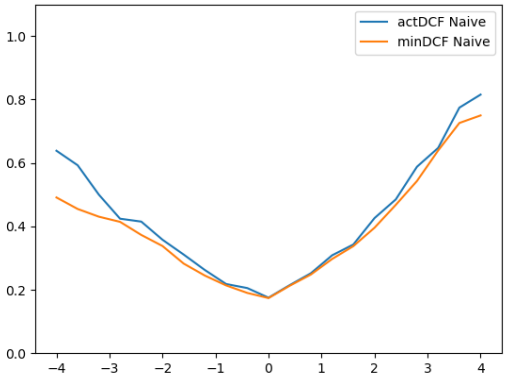

Figure 30: Naive Bayes - Bayes Error Plot

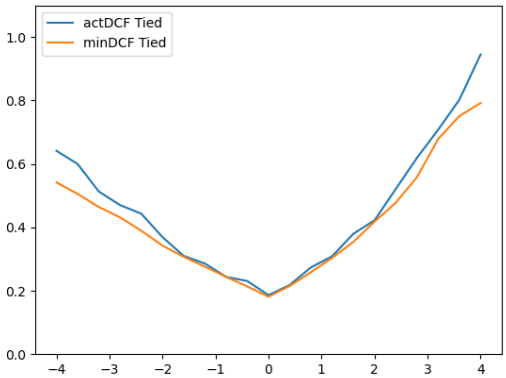

Figure 31: Tied Gaussian - Bayes Error Plot

7 Logistic Regression - Laboratory 8

In this section follows the performance of the binary logistic regression model applied to the dataset. The first step consists in evaluating the standard, non-weighted version of the model without any pre-processing steps. The model was trained with various values of the regularization parameter λ (spanning the range given by `numpy.logspace(-4, 2, 13)`), and the validation set was scored accordingly. The corresponding actual DCF and minimum DCF values were then computed for the primary application scenario where $\tilde{\pi} = 0.1$.

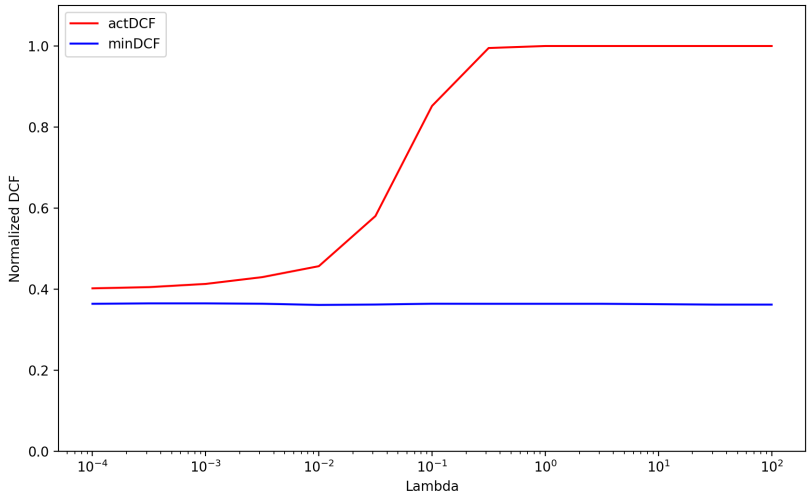

Figure 32: Logistic Regression - Bayes Error Plot for varying λ values.

Figure 32 illustrates the effect of varying the regularization parameter λ on the normalized Detection Cost Function (DCF) for a binary logistic regression model. The plot reveals that, for lower values of λ, the actual DCF (actDCF) and the minimum DCF (minDCF) are closely aligned, suggesting good model calibration and optimal decision boundaries. However, as λ increases beyond 10^{-2}, the actDCF begins to rise sharply while the minDCF remains constant. This growing gap indicates that stronger regularization adversely affects the model's performance by degrading the calibration of its scores, ultimately leading to a higher actual decision cost than the minimum achievable cost.

7.1 Comparison of Weighted and Non-Weighted Logistic Regression

For this specific application ($\tilde{\pi} = 0.1$), prior-weighted logistic regression offers minimal improvement over the non-weighted approach. Prior-weighted logistic regression may not significantly improve results if the baseline model is already well-calibrated or if the class imbalance is not severe. In such cases, the model naturally finds a good decision boundary, making the reweighting unnecessary. The minimum DCF remains low because, even with miscalibrated scores, there is always some threshold that effectively separates the classes. Additionally, if the feature space provides good separation, prior-weighting does not meaningfully shift decision boundaries.

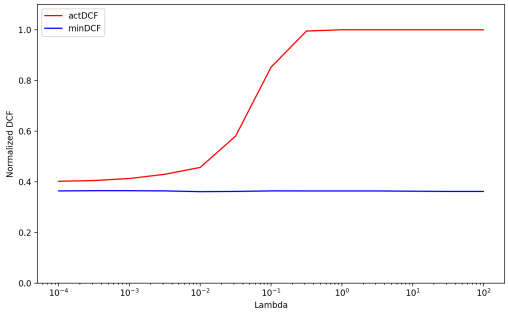

Figure 33: Non-Weighted Logistic Regression - Bayes Error Plot

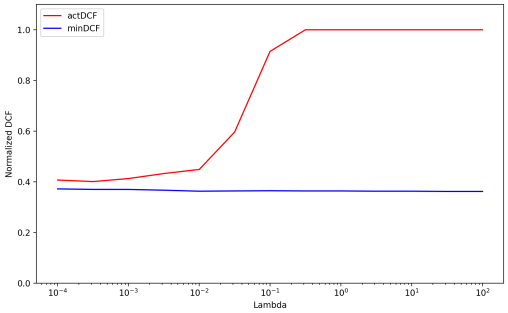

Figure 34: Weighted Logistic Regression - Bayes Error Plot

7.2 Quadratic Logistic Regression with Feature Expansion

Feature expansion was achieved by incorporating squared terms and cross-products between features, allowing the model to capture more intricate interactions. This enhancement resulted in better classification performance, with a minimum DCF of 0.2436 at $\lambda = 0.03162$. As illustrated by the plot, the expanded model initially surpasses the simpler version, but for larger λ values, the actual DCF increases significantly, signaling over-regularization.

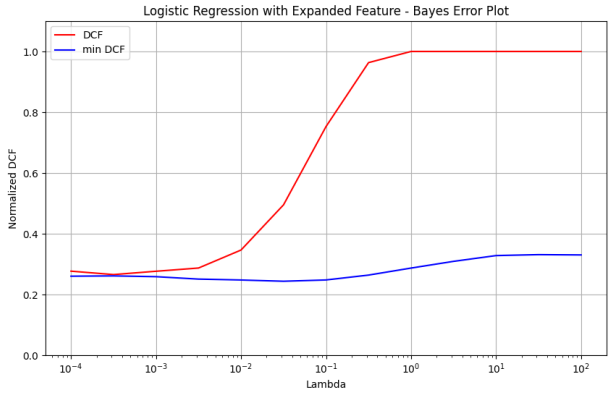

Figure 35: Bayes Error Plot for Quadratic Logistic Regression with Expanded Features.

In summary, feature expansion improves model performance with smaller λ values by enabling the model to capture more complex patterns. However, it also heightens sensitivity to regularization, resulting in underfitting when larger λ values are used.

7.3 Impact of Pre-processing Techniques

The final analysis assessed the impact of pre-processing techniques, such as PCA, centering, and Z-normalization on the performance of logistic regression. The findings revealed only minimal improvements in DCF, suggesting that the original features were already well-conditioned and that additional pre-processing had a limited effect.

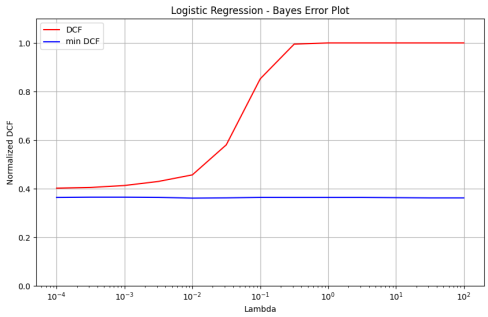

Figure 36: Non-Weighted Logistic Regression - Bayes Error Plot

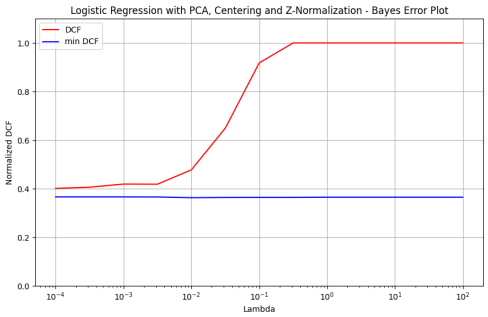

Figure 37: Weighted Logistic Regression - Bayes Error Plot

7.4 Top Performing Models

The best performance was achieved by the logistic regression model with feature expansion at $\lambda = 0.03162$, which resulted in a minimum DCF of 0.2436.

In conclusion, regularization, feature expansion, and pre-processing are key factors in optimizing logistic regression performance. Tuning λ appropriately is crucial, particularly when capturing non-linear relationships in the data.

8 Support Vector Machine - Laboratory 9

This section examines how varying the regularization coefficient C affects the performance of linear and polynomial kernel SVM models applied to our dataset. The SVM was trained using C values ranging from 10^{-5} to 10^{0} on a logarithmic scale. The parameter C is a regularization hyperparameter that controls the trade-off between maximizing the margin and minimizing classification errors (slack variables). We computed both the minimum DCF (minDCF) and the actual DCF (actDCF) for the target prior $\tilde{\pi} = 0.1$. The results are illustrated in the following visualizations.

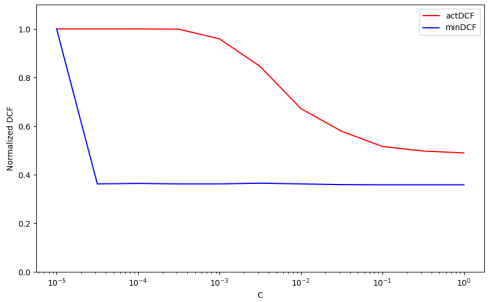

Figure 38: SVM Linear - Error Rate Analysis

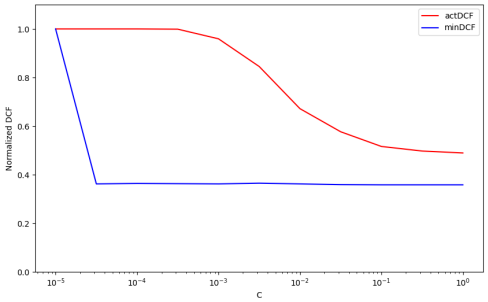

Figure 39: SVM Linear - Error Rate Analysis (Centered Data)

The regularization coefficient C significantly influences both minDCF and actDCF. At low C values (indicating strong regularization), the actDCF remains consistently high, while the minDCF decreases rapidly. This suggests that while the model effectively reduces empirical risk (as indicated by minDCF), the actual DCF is not optimized. As C

increases (less regularization), both metrics begin to stabilize; however, actDCF continues to be higher than minDCF, indicating that score calibration issues persist across all levels of regularization.

The persistent gap between minDCF and actDCF emphasizes a calibration problem with the linear SVM model. Although the model successfully establishes effective decision boundaries for minimizing empirical risk, its scores are not well-callibrated for practical use, particularly at lower C values. While the linear SVM performs comparably to other linear models in defining decision boundaries, it needs improved calibration to align actDCF more closely with minDCF.

Replicating the analysis with centralized data reveals minimal variations compared to the non-normalized version. The trends in both minDCF and actDCF remain largely consistent, as evident in the second visualization. This implies that normalizing the data does not significantly alter the linear SVM's performance, indicating the model's resilience to this transformation.

8.1 Evaluation of SVM with Quadratic Kernel

We now examine the SVM with a quadratic kernel (d = 2, c = 1, and $\xi = 0$). The following plot illustrates the results for varying C values, with both minDCF and actDCF computed for $\tilde{\pi} = 0.1$.

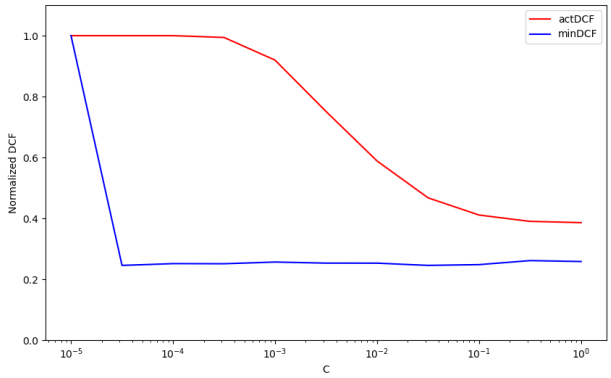

Figure 40: SVM with Quadratic Kernel - Error Rate Analysis

For small C values, both minDCF and actDCF start high, reflecting the impact of strong regularization. As C increases, minDCF decreases and stabilizes more rapidly than actDCF, which exhibits more fluctuation before settling. This gap between minDCF and actDCF suggests calibration issues, akin to those observed with the linear SVM. Nevertheless, the quadratic kernel enables the model to capture higher-order feature interactions, as reflected in the improving minimum DCF.

Quadratic Model Performance Analysis: The d=2 polynomial kernel captures more intricate decision boundaries compared to the linear SVM, but the model still exhibits calibration issues. Higher C values lead to improved minDCF, but actDCF remains inconsistent due to score miscalibration.

8.2 Assessment of SVM with RBF Kernel and Varying Regularization Parameters

In this segment, we investigate the effect of different regularization parameter C values and kernel parameter γ on the performance of the SVM model using an RBF kernel. The RBF kernel allows the SVM to model non-linear decision boundaries, which is especially useful for datasets where linear separability is inadequate. The model was trained using C values ranging from 10^{-5} to 10^{0}, combined with four different γ values (specifically, $\gamma = e^{-1}$, $\gamma = e^{-2}$, $\gamma = e^{-3}$, and $\gamma = e^{-4}$). Both the minimum DCF (minDCF) and the actual DCF (actDCF) were computed for each parameter combination.

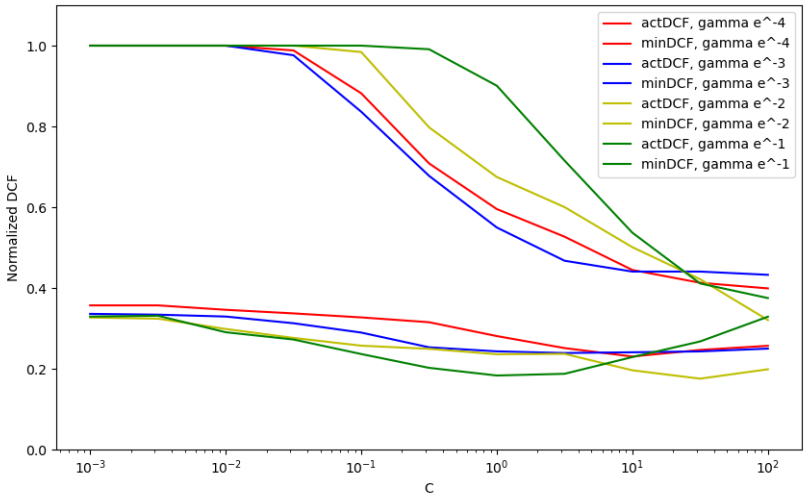

Figure 41: RBF Kernel Results

Based on the plot, it is evident that the values of γ and C significantly impact both minDCF and actDCF. Lower values of C (stronger regularization) maintain higher actDCF across all values of γ, indicating that while the model effectively minimizes the empirical risk (as observed by the decreasing minDCF), the actual performance in terms of the decision cost function remains suboptimal. As C increases, both minDCF and actDCF decrease, suggesting better model performance and more stable decision boundaries. However, even at higher values of C, a noticeable gap between minDCF and actDCF persists, particularly for lower γ values, indicating persistent calibration issues.

Certain combinations of γ and C yield better results; for instance, higher γ values (like e^{-1}) with moderate C values provide a better alignment between minDCF and actDCF. This suggests that the RBF kernel's capacity to capture complex relationships in the data might be better exploited at these parameter settings. Compared to previous linear models, the RBF kernel shows promise in capturing non-linear patterns within the dataset, which may not be well-represented by linear models. However, the persistent gap between minDCF and actDCF across most values suggests a need for further calibration to ensure optimal performance.

When the regularization coefficient C is large, the SVM model becomes less regularized, allowing it to fit the training data more closely. While this might result in a very low empirical risk (as captured by minDCF on the training data), the model may start capturing noise and specific patterns that are not generalizable to new, unseen data. This

overfitting causes the model to perform poorly on the validation or test data, which is reflected in the increasing minDCF at high C values. Therefore, the increase in minDCF at high C values suggests that the model's decision boundaries, while still minimizing the empirical risk on the training data, are becoming less effective for the actual application, indicating an overfitting issue.

8.3 Analysis of RBF and Polynomial Kernels

8.3.1 Radial Basis Function (RBF) Kernel

The Radial Basis Function (RBF) kernel is defined as:

$$K(x_i, x_j) = \exp\left(-\gamma||x_i - x_j||^2\right) \tag{1}$$

where γ is a hyperparameter that controls the spread of the Gaussian function. The RBF kernel maps the input data into an infinite-dimensional feature space, effectively capturing highly non-linear relationships. It is particularly useful for data with complex boundaries and enables support vector machines (SVMs) to adapt to intricate decision surfaces.

Effect of the γ Parameter The parameter γ plays a crucial role in determining the influence of training samples in defining the decision boundary. Specifically:

- A **small** γ value leads to a *large* variance of the Gaussian function, meaning that the kernel function has a wider spread. This results in a smoother decision boundary, as distant data points have a stronger influence on classification. However, setting γ too low may lead to underfitting, where the model fails to capture the complexity of the data.
- A **large** γ value leads to a *small* variance of the Gaussian function, meaning that the kernel function has a narrow spread. This makes the model focus more on the immediate neighborhood of each training point, creating highly localized decision boundaries. While this can increase model flexibility, excessively high γ values can lead to overfitting, where the model memorizes the training data rather than generalizing well to new data.

8.3.2 Polynomial Kernel

The polynomial kernel is given by:

$$K(x_i, x_j) = (x_i^T x_j + c)^d \tag{2}$$

where d is the polynomial degree, and c is a coefficient that controls the influence of higher-order terms. The polynomial kernel maps data into a finite-dimensional feature space where higher-order interactions between features are explicitly captured. This makes it effective for moderately non-linear problems while maintaining interpretability.

8.3.3 Why Do RBF and Polynomial Kernels Perform Similarly?

In our binary classification task of fingerprint spoof detection, both the RBF and polynomial kernels achieve comparable performance. This similarity arises due to several factors:

- **Feature Dimensionality:** Our dataset consists of six extracted features. Given this relatively low-dimensional feature space, both kernels can effectively capture the relationships without requiring an extremely complex mapping.
- **Class Separation:** The feature distributions indicate that while some features provide distinct separation, others exhibit overlap. Both kernels can model these separations effectively, leading to similar decision boundaries.

8.3.4 Expected Behavior of Each Kernel

RBF Kernel: We expect the RBF kernel to perform well in scenarios where the decision boundary is highly non-linear and requires a flexible model to capture intricate variations. It is generally more robust to overfitting compared to high-degree polynomial kernels when γ is properly tuned.

Polynomial Kernel: The polynomial kernel is expected to work well when the decision boundary can be approximated by a polynomial function of moderate degree. It may be more interpretable than RBF, but high-degree polynomials can lead to overfitting, particularly if the dataset contains noise.

9 Gaussian Mixture Models (GMMs) - Laboratory 10

In this section, we assess the performance of Gaussian Mixture Models (GMMs) with different covariance structures—full, diagonal, and tied covariance—across varying numbers of components per class. Our objective is to identify the optimal model configuration by evaluating its performance on the validation set. We analyze minimum Detection Cost Function (minDCF) to determine which model best suits our target application. By comparing these covariance structures, we aim to understand their impact on classification performance and select the most effective approach for our use case.

Model	**Components (Class 0)**	**Components (Class 1)**	**Minimum DCF**
Full GMM	8	16	0.154
Diagonal GMM	4	16	0.136
Tied GMM	1	8	0.155

Table 13: Comparison of Full, Diagonal, and Tied GMM models.

9.1 Analysis of the results

Based on the results presented in Table 13, we observe that the Diagonal GMM achieved the lowest minimum DCF (0.136), outperforming both the Full GMM (0.154) and the Tied GMM (0.155). This performance difference can be attributed to the flexibility and regularization effects of each covariance structure. The Full GMM, while offering the highest flexibility by modeling full covariance matrices, can suffer from overfitting, especially when the number of components is relatively high compared to the available data. This could explain its slightly weaker performance compared to the Diagonal GMM. The Diagonal GMM, which assumes independence between features, provides a good balance

between flexibility and regularization, avoiding overfitting while still capturing relevant class distributions. The Tied GMM, on the other hand, enforces a shared covariance matrix across all mixture components, which may overly constrain the model's ability to adapt to class-specific variations, leading to a slightly higher DCF.

Furthermore, in Chapter 5, we analyzed the correlation matrix of the feature set and observed that the features were not highly correlated. This finding further supports the strong performance of the Diagonal GMM, as the assumption of feature independence is reasonably valid in this case. Since the off-diagonal elements of the correlation matrix were relatively low, enforcing a full covariance structure did not provide significant advantages and may have introduced unnecessary complexity, leading to overfitting. On the other hand, the Tied GMM's shared covariance assumption might have been too restrictive, limiting its ability to model class-specific distributions effectively. Thus, the results reinforce the idea that in this scenario, the Diagonal GMM is the most effective choice, as it successfully captures the class distributions while maintaining a good balance between complexity and generalization.

9.2 Models performance comparison

Model	**Minimum DCF**
MVG	0.263
MVG Naïve Bayes	0.257
MVG Tied	0.361
Logistic Regression	0.360
Logistic Regression (Expanded features)	**0.244**
SVM (Linear)	0.358
SVM (Polynomial Kernel)	**0.174**
SVM (RBF Kernel)	**0.175**
GMM (Full, 8, 16)	0.154
GMM (Diagonal, 4, 16)	**0.136**
GMM (Tied, 1, 8)	0.155

Table 14: Performance Comparison of Different Models

9.2.1 Performance Comparison of Classification Models

Table 14 presents a comparison of different classification models based on their Minimum Detection Cost Function (DCF). The results reveal key insights into the strengths and weaknesses of various approaches.

Best Performing Models The best models in terms of Minimum DCF are:

- **GMM (Diagonal, 4, 16)**: **0.136** (best overall performance)
- **SVM (Polynomial Kernel)**: 0.174 (best SVM variant)
- **SVM (RBF Kernel)**: 0.175 (second-best SVM)

- **Logistic Regression (Expanded Features)**: 0.244 (best performing logistic regression model)

These models achieve superior performance by capturing complex relationships between features, either through non-linearity (SVMs, expanded logistic regression) or flexible probabilistic modeling (GMM).

Gaussian Mixture Models (GMM) The GMM models exhibit strong performance, particularly the **Diagonal GMM (0.136)**, which outperforms all others. The **Full GMM (0.154)** performs slightly worse, likely due to overfitting, while the **Tied GMM (0.155)** shows lower adaptability due to its shared covariance assumption. Since the dataset's features exhibit low correlation, the diagonal covariance assumption aligns well with the data.

Support Vector Machines (SVM) The results highlight the importance of using non-linear kernels:

- **Polynomial Kernel** (0.174) performs best among SVMs.
- **RBF Kernel** (0.175) follows closely, showing strong performance in capturing non-linear boundaries.
- **Linear SVM** (0.358) performs poorly, indicating that the dataset is not linearly separable.

The significant improvement from linear to non-linear SVMs confirms that the dataset requires more complex decision boundaries.

Logistic Regression A similar trend is observed in logistic regression:

- **Basic Logistic Regression** (0.360) performs poorly.
- **Logistic Regression with Feature Expansion** (0.244) improves significantly by incorporating squared terms and interactions, demonstrating that higher-order relationships exist in the data.

Multivariate Gaussian Models (MVG) The MVG models exhibit moderate performance:

- **MVG** (0.263) shows a reasonable classification capability.
- **Naïve Bayes MVG** (0.257) improves slightly, suggesting that assuming feature independence does not significantly harm performance, it actually improves it.
- **Tied MVG** (0.361) performs the worst, likely due to its restrictive covariance assumptions.

Key Observations

- **Feature Independence Assumption:** The decent performance of Naïve Bayes MVG suggests that feature independence is not a strong limitation in this dataset.
- **Non-Linearity Matters:** Linear models (Linear SVM, Basic Logistic Regression) fail, showing that the dataset requires more complex, non-linear classifiers.
- **GMM vs. SVM:** The best-performing GMM (0.136) outperforms the best SVM (0.174), indicating that probabilistic modeling is more effective in this case.

9.3 Performance for Different Applications

Figure 42 illustrates the Bayes error plot for the selected models, comparing their Detection Cost Function (DCF) and minimum DCF across different prior probabilities. The analysis highlights the behavior of Quadratic Logistic Regression, Polynomial SVM (degree 4), RBF SVM, and GMM with Diagonal covariance.

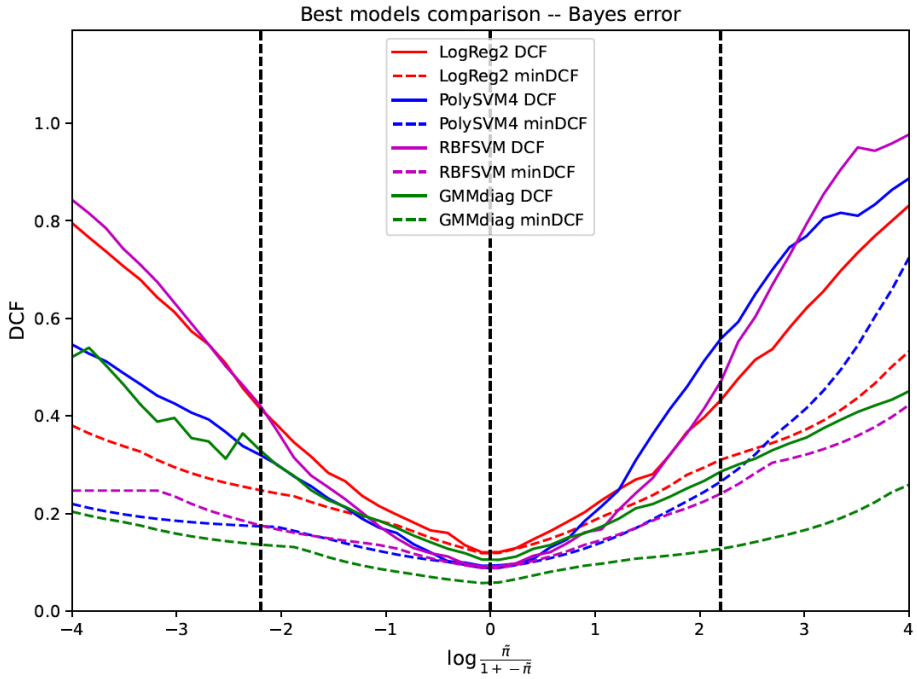

Figure 42: Bayes error comparison of the best-performing models. The plot shows the DCF and minDCF curves for Quadratic Logistic Regression, Polynomial SVM, RBF SVM, and GMM with Diagonal covariance.

The Bayes error plot highlights both classification performance and model calibration by comparing DCF and minDCF across different prior probabilities. GMM with Diagonal covariance (GMMdiag) emerges as the best-performing model, consistently achieving the lowest DCF while maintaining a minimal gap between DCF and minDCF, indicating strong calibration and reliable probability estimates. SVM models (PolySVM4 and RBF SVM) perform competitively, but their calibration deteriorates at extreme prior values, as evidenced by the widening gap between DCF and minDCF. In contrast, Quadratic Logistic Regression (LogReg2) shows significant miscalibration, with a consistently large gap between its DCF and minDCF, suggesting poor probability estimation and suboptimal

decision-making. Overall, GMMdiag stands out as the most robust and well-calibrated model, while SVM models remain viable but less reliable under class imbalances, and LogReg2 struggles due to miscalibration.

10 Calibration and Fusion - Laboratory 11

The most effective model in this study is the Gaussian Mixture Model (GMM) with a diagonal covariance matrix, as it achieves the lowest minimum Decision Cost Function (DCF). Given this, we investigate the impact of calibrating its scores to further enhance results.

10.1 Performance Comparison on the Validation Set

Table 15 presents the comparative performance of various models following calibration.

Table 15: Results of K-fold calibration on the validation set

Model	actDCF	minDCF
Quadratic LR	0.270	0.244
Polynomial SVM	0.187	0.174
RBF SVM	0.189	0.175
GMM Diagonal	0.166	0.136

Figure 43 illustrates the Bayes error plot for the top-performing calibrated models.

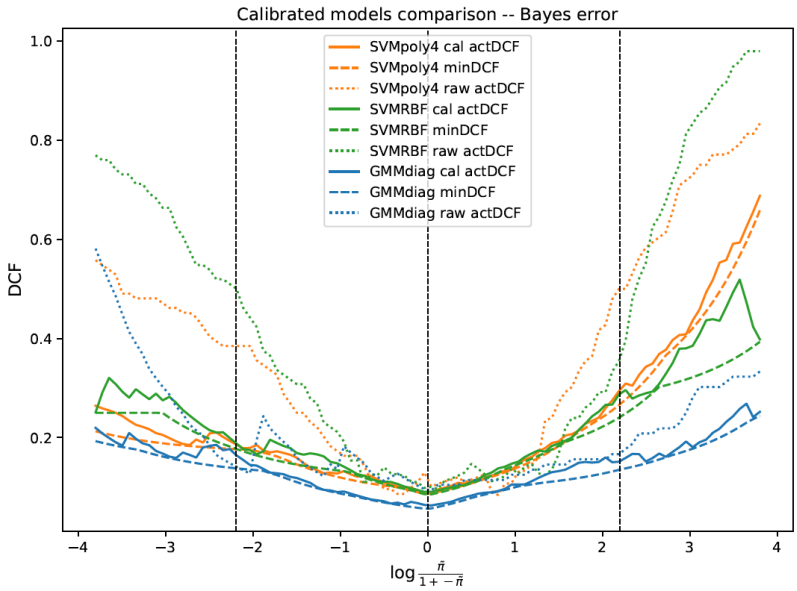

Figure 43: Bayes error comparison of calibrated models on the validation set.

11 Fusion of Models

A promising approach to further enhance performance is model fusion. By integrating the strengths of multiple models, we can leverage their unique contributions to improve

detection capabilities.

11.1 Performance of the Fused Model

The delivered model is the fused version of the three models: Quadratic Logistic Regression, RBF Support Vector Machine and GMM with diagonal covariance matrix.

Table 16 summarizes the effectiveness of the fused model, which achieves the lowest actual DCF (0.160), establishing it as the final model for this work.

Table 16: Performance of the fused model on the validation set

actDCF	**minDCF**
0.160	0.128

Figure 44 showcases the Bayes error performance of the fused model.

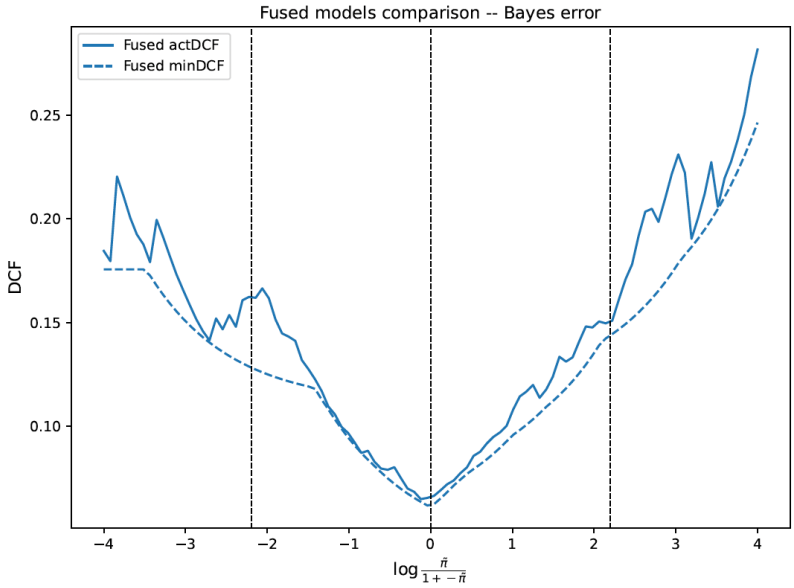

Figure 44: Bayes error comparison of fused models on the validation set.

12 Final Evaluation and Conclusion

The final model combines Quadratic Logistic Regression, RBF Support Vector Machine, and GMM with a diagonal covariance matrix. Table 17 presents the performance on the evaluation set, where the fused model demonstrates superior reliability by balancing the strengths of individual models.

Figure 45 presents the final Bayes error comparison across calibrated models.

Overall, the fusion of models provides a more robust framework for our security application, reinforcing the effectiveness of model combination in reducing errors and enhancing performance.

Table 17: Final model evaluation results

Model	**actDCF**	**minDCF**
Quadratic LR	0.359	0.351
Polynomial SVM (Degree 4)	0.277	0.254
RBF SVM	0.289	0.262
GMM Diagonal	0.197	0.178
Fused Model	0.196	0.193

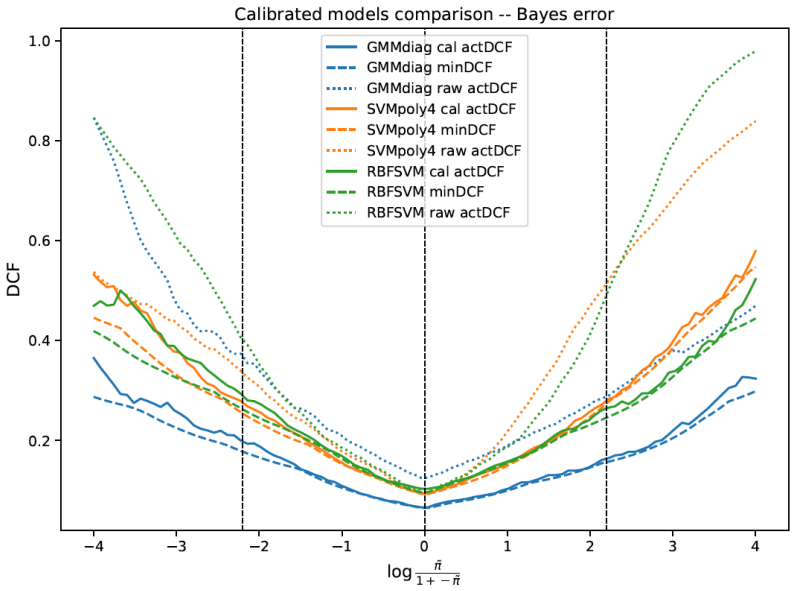

Figure 45: Bayes error plot for calibrated models on the evaluation set.