Machine Learning & Pattern Recognition Final Project Report

Can Karacomak - 287864
Politecnico di Torino
Corso Duca degli Abruzzi, 24, 10129 Torino TO

June 2022

1 Data Set Information and Introduction

The Gender Detection dataset is made up of synthetic speaker embeddings that represent the acoustic properties of spoken pronunciation. There are 12 features in each row, followed by the gender designation, and each row represents a different speaker (1 for female, 0 for male). There is no specific interpretation for the features. Four different age groups of speakers are present. However, there is no information accessible on age.

2000 samples per class are in the test set whereas 3000 samples per class are in the training set.

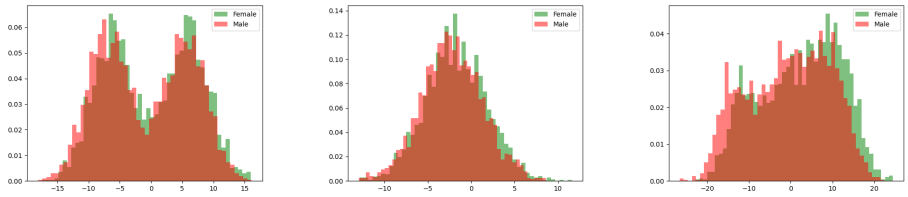
Single fold with splitting data 4 to 1 will be adopted for all the experiments. Because of the time limitation, a single fold is adopted. However, even if k-fold cross validation has not been used, there is some implementation of k-fold cross-validation in the source code. This situation may cause weak decisions. But it will be seen in the experiment.

Note: The final model will be a fusion of the best two models. However, in a scenario that if only one model has to be delivered single-fold validation can cause weak decisions. In this scenario, the final decision will be made by k-fold cross-validation with k=5 from the best two models. This comparison will be made only for these two best models. There will not be any comparison between single-fold and k-fold cross-validation results.

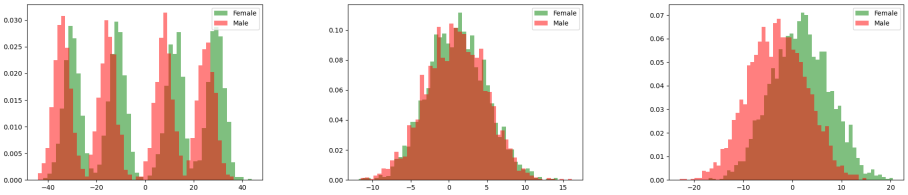
Moreover, the dataset has exactly balanced samples. Therefore, any re-balanced application would be useless. However, just for educational purposes, the re-balanced applications will have experimented with in the SVM models.

1.1 Feature Analysis

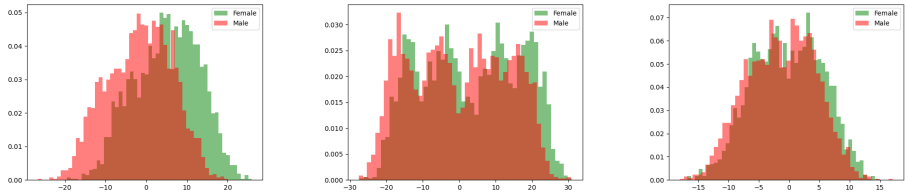
As it can be seen in figure 1, the distribution of the raw data is quite wide and similar for both classes. There are no noises that we need to consider. It may be said that the dataset is naturally suitable for the classification task. Some further pre-processing on the dataset may not be needed.



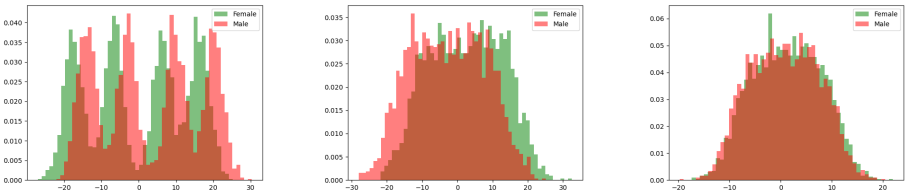
(a) Feature 1 (b) Feature 2 (c) Feature 3



(d) Feature 4 (e) Feature 5 (f) Feature 6



(g) Feature 7 (h) Feature 8 (i) Feature 9



(j) Feature 10 (k) Feature 11 (l) Feature 12

Figure 1: Distribution of raw features.

Especially, it would be necessary to mention that features 4 and 10 are nested themselves about class distributions. The models such as the Gaussian Mixture Model (GMM) can overcome and may even make this kind of challenge a benefit with their technique. Because these features have different aspects (means are different for classes) between classes as they can be observable. This will be touched on the future experiments.

Additionally, features 6, 7 and relatively 11 will be helpful for the separation rules since these features have differences for classes. And since the distributions are similar and means are different for classes, gaussian classifiers with a tied covariance matrix should perform well.

As it can be observed in figure 2, gaussiniazer is not helpful in this case. The features are not better distributed since their variances were wide and there were no noises already. It would be said that some information may have disappeared in features 4 and 10. Especially, this may affect the GMM classifier. Also, this will have experimented in the GMM experiment part.

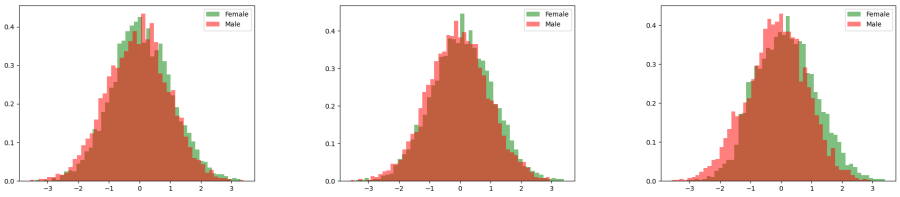
A correlation analysis 3 and 4 shows that the features are highly correlated. And it can be said that the classifiers that has the independent feature assumption may be perform bad in this dataset such as the Multivariate Gaussian classifiers with the naive covariance matrix (Dimension reduction may help to reduce correlation in the new feature space.).

About dimension reduction, since the features are correlated and this correlation in both class, the dimension reduction techniques would keep the information in data as much as possible. Nevertheless, there are only 12 dimension. There is not the curse of dimensionality. There are not unwanted variability since there are not noises. We will be experimented that dimension reduction will effect better or worse with this dilemma. Especially we will observe in evaluation data (the validation data may be too similar with training data) that, the dimension reduction will cause simplified classification and help about over-fitting or cause information loss. It can be said that dimension reduction may not help essentially to the sophisticated quadratic models such as GMM or SVM with kernel trick. However, it should help linear and/or more simple models such as MVG.

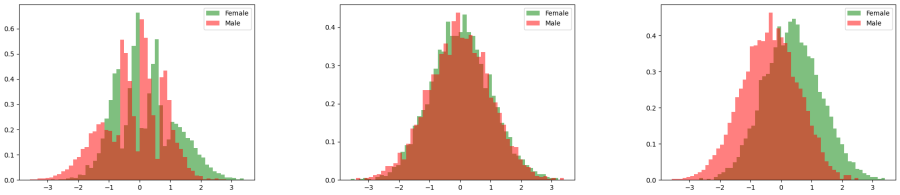
2 Dimension Reduction

As it can be seen in figure 5, PCA shows the natural distribution of the dataset in two-dimensional feature space. And we can say that GMM and SVM with kernel trick may reach good results even with a few features in PCA dimension reduction. Meanwhile, LDA shows that even simple classifiers such as MVG and linear classifiers may reach good results with a feature in LDA dimension reduction since the dataset is already well separated.

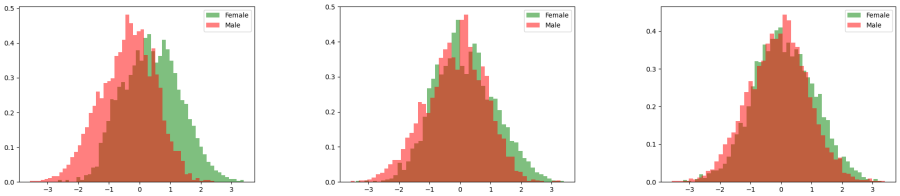
Furthermore, dimension reduction analysis will be made in the each classifiers to determine best model since simple decision rules can produce more general separation rules. And the results will have experimented especially in the evaluation dataset.



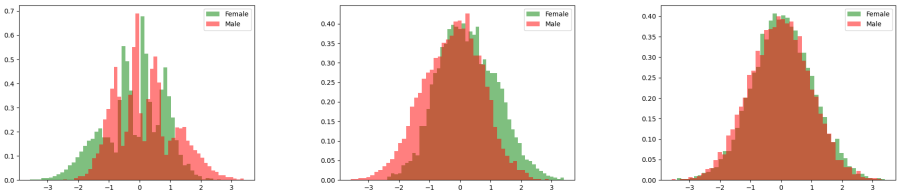
(a) Feature 1 (b) Feature 2 (c) Feature 3



(d) Feature 4 (e) Feature 5 (f) Feature 6

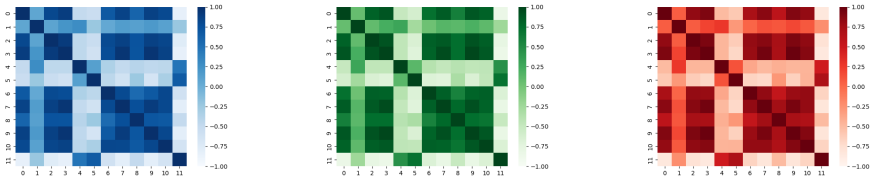


(g) Feature 7 (h) Feature 8 (i) Feature 9



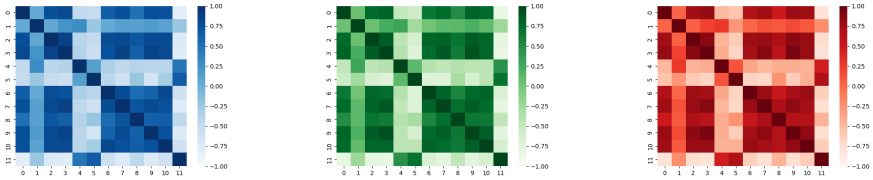
(j) Feature 10 (k) Feature 11 (l) Feature 12

Figure 2: Distribution of gaussiniazed features.



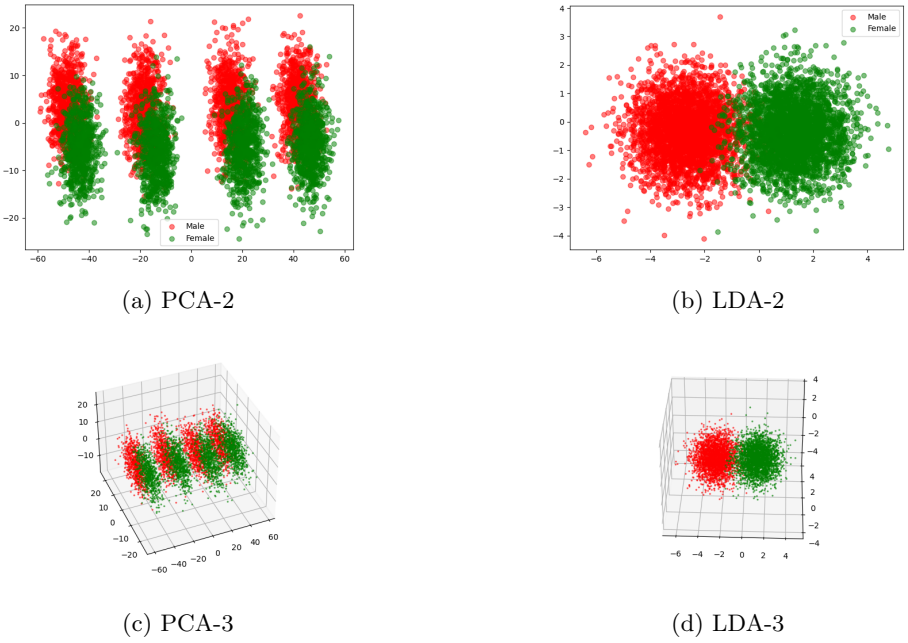
(a) All data set (b) Female class (c) Male class

Figure 3: Correlation heat-map of features on raw data



(a) All data set (b) Female class (c) Male class

Figure 4: Correlation heat-map of features on gaussiniazed data



(a) PCA-2 (b) LDA-2

(c) PCA-3 (d) LDA-3

Figure 5: Scatter plots of the dataset with indicated dimension reduction technique

Note: The figures 5b and 5d are showed for visualization purpose. Linear Discriminant Analysis can be used with at most number of class minus one dimensions. In our binary case, we can use with 1 dimension. And it does not make sense to have good result in our case. Nevertheless, in MVG section, it will be experimented just for education purpose.

3 Classification

In order, Multivariate Gaussian Classifier, Logistic Regression, Support Vector Machine and Gaussian Mixture Model will be applied to validation data to find best classifier on the data set. The methodology is the training data is separated to two set which are training (%80) and validation (%20) sub-sets. And these sub-sets will be used for all the classification technique. To have the same results, shuffle seed is the same in each.

Main models will be decided with the following prior probability for positive class, and the costs for each class.

$$(\pi, C_{fp}, C_{fn}) = (0.5, 1, 1) \tag{1}$$

However, also unbalanced applications will be considered with following indications.

$$(\pi, C_{fp}, C_{fn}) = (0.1, 1, 1), (\pi, C_{fp}, C_{fn}) = (0.9, 1, 1) \tag{2}$$

Since it is better to perform the selection with trustworthy way. Minimum detection cost is adopted in this report. This technique measures the cost that perform the classification if the optimal decision has been made with the optimum threshold. The minimum detection cost shows the potential of the model.

3.1 Validation

3.1.1 Gaussian Classifier

Our first model is a generative model. Basically, generative classifiers uses Bayes rules to calculate posterior probability by empirical information such as mean and variance and additionally prior probability that is chosen. And they produce a probability for each sample in target set.

As we can see in the table 1; the best result has been received by the model that has full covariance matrix and LDA-1 dimension reduction on raw data. It can be said that, since the data is suitable for a simple separation rule, the model could perform with just 1 dimension. And the results are exactly the same in all kind if covariance matrices. Since there is 1 dimension which means a dot represents the separation rule, it is understandable.

However, more complicated models must perform better with high number of features. Because the results are close comparison to the model that has full covariance matrix without dimension reduction. And as we can see in figure 5a,

Table 1: MVG - Minimum Detection cost on Validation Set

	Single Fold					
	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Model	**Gaussianized**			**Raw Data**		
Full-Cov	0.065	0.228	0.172	0.061	0.148	0.129
Naive-Cov	0.527	0.848	0.809	0.55	0.854	0.806
Tied-Cov	0.063	0.231	0.159	0.0617	0.138	0.116
Naive-Tied-Cov	0.547	0.842	0.79	0.568	0.849	0.831
PCA-11						
6 Full-Cov	0.08	0.22	0.214	0.061	0.148	0.12
Naive-Cov	0.1	0.288	0.221	0.086	0.235	0.183
Tied-Cov	0.081	0.244	0.203	0.063	0.139	0.123
Naive-Tied-Cov	0.081	0.239	0.204	0.063	0.143	0.122
PCA-10						
Full-Cov	0.078	0.225	0.213	0.061	0.157	0.116
Naive-Cov	0.099	0.283	0.211	0.083	0.223	0.181
Tied-Cov	0.083	0.248	0.204	0.063	0.148	0.118
Naive-Tied-Cov	0.083	0.241	0.206	0.061	0.154	0.127
LDA-1						
Full-Cov	0.061	0.228	0.156	0.058	0.139	0.113

MVG classifiers are not the best match with this kind of distributions. It can be improved by the SVM or the GMM classifiers.

As we expected, the minimum detection costs increase in gaussianized features. Therefore, it will not be analyzed deeply in this report since there is no good effect of it on results.

For the naive models, dimension reduction helps significantly. This situation may explain with more independent feature space. PCA would decrease the correlation in the new feature space. And this situation affects naive models more than others.

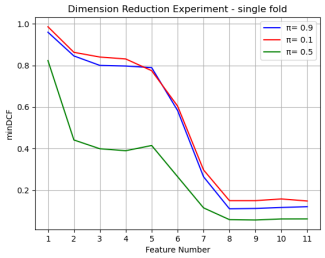
Additionally, the minimum detection cost decreases slightly with the dimension reduction PCA in each step. In the figures 6, it can be seen the effects of PCA with more details. The PCA analysis are made on raw features with the indicated covariance matrices.

As it can be observable in the figure 6, all the information are kept until 8 dimension remains. With fewer dimensions, the information loss is starting.

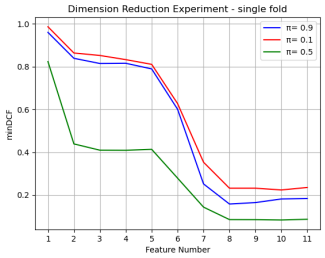
Overall, best models with their minimum detection costs as in the following table 2.

Table 2: MVG - Minimum Detection cost on Validation Set - The Bests

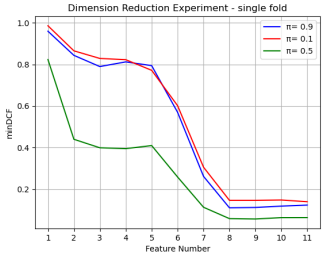
	Single Fold		
	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Model	**Raw**		
PCA-9			
MVG (Full-Cov)	0.056	0.149	0.112
MVG (Tied-Cov)	0.056	0.146	0.112
LDA-1			
MVG (Full-Cov)	0.058	0.139	0.113



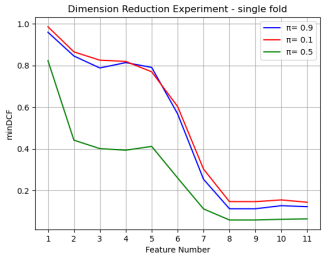
(a) with full covariance matrix



(b) with naive covariance matrix



(c) with tied covariance matrix

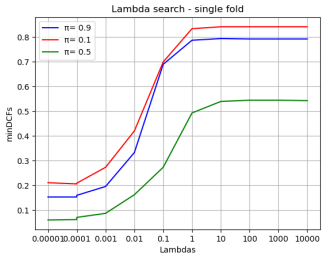


(d) with naive tied covariance matrix

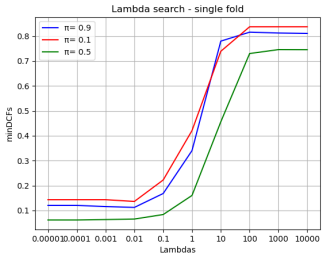
Figure 6: Detailed dimension reduction analysis for every possible reduction in MVG models

3.1.2 Linear Logistic Regression

Our second model is a discriminative model. Basically, discriminative classifiers uses a empirical hyperplane that separate the classes. And classification is performed with the position of input data according to this hyperplane. And in training, the hyperplane is positioned with penalty system that costs for each sample that miss-classified.



(a) Gaussianized data



(b) Raw data

Figure 7: Lambda search with single fold

Figure 7a is for showing the results are not better on the gaussianized features. Nevertheless, we can achieve similar results in balanced application. It cannot be said for in balanced application.

As it can be seen in the figure 7b, small lambdas affects better than high choices. However, the results do not change significantly smaller than 0.01. Lambda 10^{-4} is selected for the logistic regression model.

Since the dimension reduction affects the results significantly, considering possibilities would be better. Following figure 8 shows us the result of the dimension reduction experiment on the logistic regression classifier.

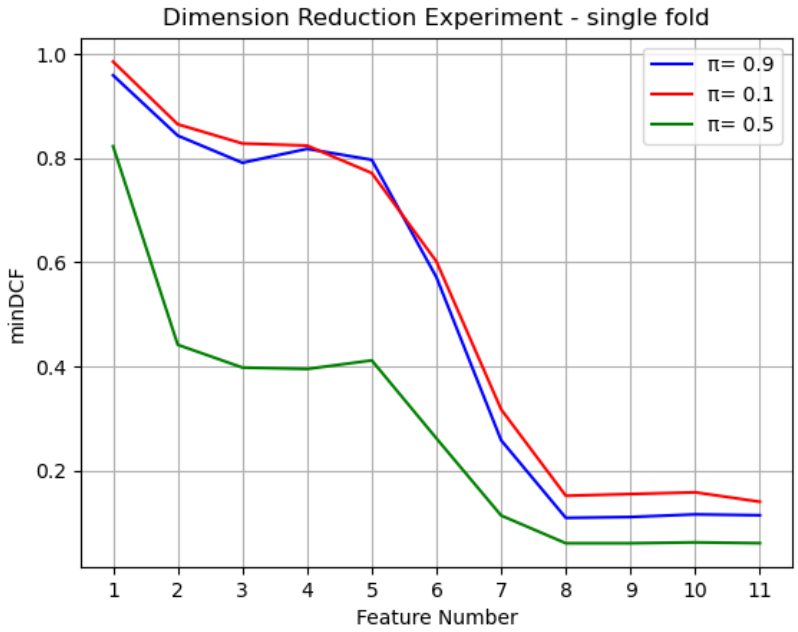


Figure 8: Detailed dimension reduction analysis for every possible reduction on Logistic Regression classifier

As we can observe in the figure 8, the minimum cost decreases slightly for the unbalanced applications, while balanced application cost keeps the same from dimension 10 to 8. Also there is small rise from dimension 11 to 10. Best results can be seen in the next table 3.

As expected, results are quite similar in balanced application as it can be seen in table 3. We can achieve same or better results in unbalanced applications with logistic regression. However, MVG model that has a tied or a full covariance matrix with dimesnion reduction PCA 9 is still the best overall in the balanced application.

Table 3: LR - Minimum Detection cost on Validation Set

Single Fold			
	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Model		**Raw**	
Full Feature			
LR ($\lambda = 10^{-4}$)	0.061	0.143	0.12
PCA-11			
LR ($\lambda = 10^{-4}$)	0.06	0.139	0.113
PCA-8			
LR ($\lambda = 10^{-4}$)	0.06	0.151	0.108
PCA-9			
MVG (Full-Cov)	0.056	0.149	0.112
MVG (Tied-Cov)	0.056	0.146	0.112
LDA-1			
MVG (Full-Cov)	0.058	0.139	0.113

3.1.3 Linear Support Vector Machine

For SVM models, fine-tuning is crucial to find the best hyperparameters C and K than to reach the best performance. C affects more classification tasks. Single fold with separation 4 to 1 is adopted as in all the other experiments.

(a) Extensive hyperparameter space

(b) Narrowed hyperparameter space for easy to recognize

Figure 9: Hyperparameter tuning on raw data

As it can be seen on figure 9b, the blue bar points the best hyperparameters which are $C = 10$ and $K = 10^{-1}$.

For the table 4, the linear SVM model performed better than all the other models in balanced applications with a tiny difference. Also, the linear SVM models could achieve the same performance with PCA dimension reductions, in comparison to other models in unbalanced applications.

Each training takes a significant time. Because of that, detailed dimension reduction analysis could not be experimented with. However, the best results for each application can be found in the table 4 with the necessary dimension reduction.

Table 4: Linear SVM - Minimum Detection cost on Validation Set

Single Fold			
	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Model		**Raw**	
Full Feature			
SVM ($C = 10$ & $K = 10^{-1}$)	0.055	0.142	0.115
SVM ($C = 10$, $K = 10^{-1}$, $\pi = 0.5$)	0.063	0.149	0.125
PCA-11			
LR ($\lambda = 10^{-4}$)	0.06	0.139	0.113
SVM ($C = 10$ & $K = 10^{-1}$)	0.061	0.139	0.12
PCA-10			
SVM ($C = 10$ & $K = 10^{-1}$)	0.061	0.148	0.108
PCA-8			
LR ($\lambda = 10^{-4}$)	0.06	0.151	0.108
PCA-9			
MVG (Full-Cov)	0.056	0.149	0.112
MVG (Tied-Cov)	0.056	0.146	0.112
LDA-1			
MVG (Full-Cov)	0.058	0.139	0.113

3.1.4 Quadratic Support Vector Machine with Kernel

We will analyze two different kernel approaches in quadratic SVM. First is radial basis function (RBF) kernel 3 (based on euclidean distance between two points) and the other one is polynomial kernel 4 (based on the similarity of vectors in a feature space over polynomials of the original variables).

$$kernel(x_1, x_2) = e^{-\gamma||x_1 - x_2||^2} \tag{3}$$

$$kernel(x_1, x_2) = (x_1^T x_2 + constant)^d \tag{4}$$

For the figures 10, the bar charts show all parameter spaces that the search has made. However, Due to hard recognition, there are also line graphs with narrowed search spaces for the best hyperparameters in all search spaces.

In the graph 10b, we can see the best constant of kernel is 1 while C equals 10^{-2}. And in the graph 10d, the best C is 1 while γ equals 10^{-2}. And those are will be our hyperparameters for the quadratic SVM models.

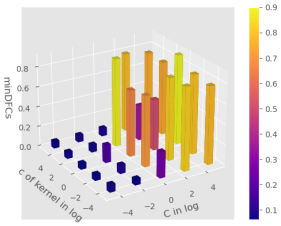
Note: K always equals 10^{-1} in all the quadratic SVM models.

Even if, our data has balanced classes. I would like to show re-balanced applications. For this, the same technique in the course is adopted. Basically, a different C are used for each classes and the C is calculated as in following equations.

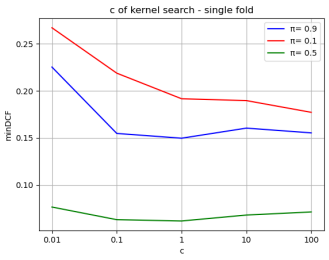
$$C_T = C\frac{\pi_T}{\pi_T^{emp}}, C_F = C\frac{\pi_F}{\pi_F^{emp}} \tag{5}$$

π_T^{emp} and π_F^{emp} are empirical priors depends of the sample ratio for classes.

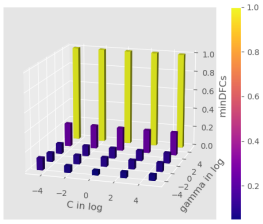
For the table 5, the RBF kernel performed significantly better than the polynomial kernel. Because of that, re-balanced applications are made with



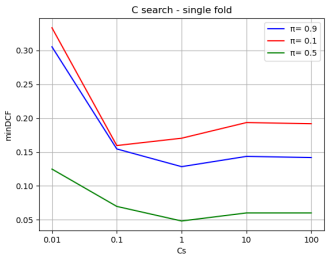
(a) particular constant and C combinations for the polynomial kernel



(b) Clear search of constant of kernel with $C = 10^{-2}$



(c) particular gamma and C combinations for the RBF kernel



(d) Clear search of C with $\gamma = 10^{-2}$

Figure 10: Hyperparameter tuning for the quadratic SVM classifiers with kernel on raw data. The top graphs are for the polynomial kernel, bottom graphs are for RBF kernel

Table 5: Kernel SVM - Minimum Detection cost on Validation Set

Single Fold			
	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Model	**Raw**		
Full Feature			
Poly Kernel SVM ($C = 10^{-2}, c = 1$)	0.061	0.191	0.149
RBF Kernel SVM ($C = 1, \log \gamma = -2$)	0.048	0.17	0.128
RBF Kernel SVM ($\pi = 0.5$)	0.048	0.17	0.12
RBF Kernel SVM ($\pi = 0.1$)	0.058	0.127	0.166
RBF Kernel SVM ($\pi = 0.9$)	0.056	0.177	0.121
Linear SVM ($C = 10$ & $K = 10^{-1}$)	0.055	0.142	0.115
PCA-10			
Linear SVM ($C = 10$ & $K = 10^{-1}$)	0.061	0.148	0.108
PCA-8			
LR ($\lambda = 10^{-4}$)	0.06	0.151	0.108

the RBF kernel. And the SVM model with the RBF kernel performed better than all the other models in the balanced application and one of the unbalanced applications with re-balancing. Significant improvement has been made by the RBF kernel.

It can be said that the SVM model with the polynomial kernel is relatively useless, in comparison with the other models.

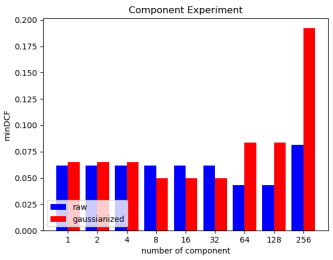
As we can see, the re-balanced application could not perform successfully. Since our data was already in balance, extra interference did not make the results better. However, we will see further experiments in the next sections.

Additionally, particular dimension reduction results have been checked but there did not find any better results for the best model. Because of that, it was not reported.

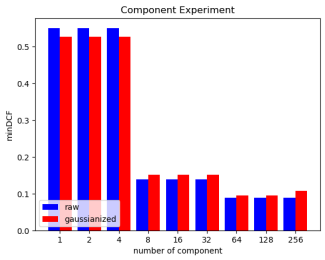
Overall, the best model is the SVM classifier with the RBF kernel that has C=1 and $\gamma = 10^{-2}$.

3.1.5 Gaussian Mixture Model

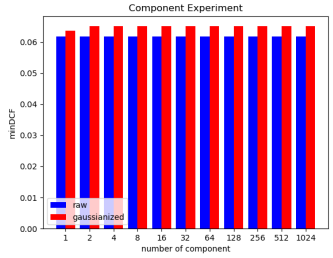
We will analyze all four of the covariance techniques with different component numbers. Since, this analyze requires computational source. Our experiment will be limited by this issue.



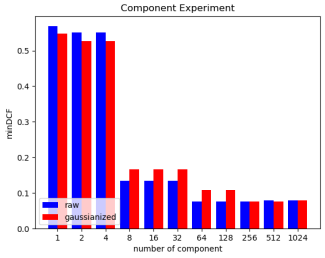
(a) Full Covarianve Matrix



(b) Naive Covarianve Matrix



(c) Tied Covarianve Matrix



(d) Naive Tied Covarianve Matrix

Figure 11: Component analysis with single fold

For the figure 11, all the models except the model with a naive tied covariance matrix performed best on raw data. For the model with a naive tied covariance

matrix, it can be performed totally the same on the raw or the gaussianized data with the different number of components.

It can be observable in figure 11a, overfitting problem would occur after 64 for the gaussianized data and after 256 for the raw data.

Overall, the GMM model that has a full covariance matrix and 64 or 128 components performed better than all the other models.

Table 6: GMM - Minimum Detection cost on Validation Set

Single Fold			
	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Model	**Gaussianized**		
RBF Kernel SVM ($C = 1, \log \gamma = -2$)	0.048	0.17	0.128
RBF Kernel SVM ($\pi = 0.5$)	0.048	0.17	0.12
RBF Kernel SVM ($\pi = 0.1$)	0.058	0.127	0.166
RBF Kernel SVM ($\pi = 0.9$)	0.056	0.177	0.121
GMM (Full Cov / 64-Raw)	0.043	0.142	0.118
GMM (Naive Cov / 256-Raw)	0.09	0.267	0.241
GMM (Tied Cov / 1024-Raw)	0.077	0.227	0.198
GMM (Naive Tied Cov / 64-Raw)	0.061	0.148	0.129

As it can be observed in table 6, full covariance model performed best against all the other models for the balanced and the unbalanced with prior equals 0.9 applications. However, the SVM with RBF kernel with rebalancing is better for the unbalanced application with prior equals 0.1

Single fold validation was used until this point. And now, further investigation will be made into the best two models by k-fold cross-validation. We will see whether the results will change or not.

Note: This investigation is for the mentioned scenario.

Table 7: Minimum Detection Costs of k-fold Cross Validation

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
SVM	0.042	0.121	0.124
GMM	0.049	0.148	0.139

As it is observable in table 7, the results are surprising. The minimum costs are totally different than expected. The SVM model performed significantly better than the GMM model in all types of applications. And these results changed the final decision. Because the k-fold cross validation should give more trustworthy results.

The primary model is the SMV model that has a RBF kernel with previously indicated hyperparameters.

The secondary model is the GMM model thath has 64 components with full convariance matrix.

3.1.6 Analysis

All the analyses will be made with k-fold validation scores. It will give more rich information.

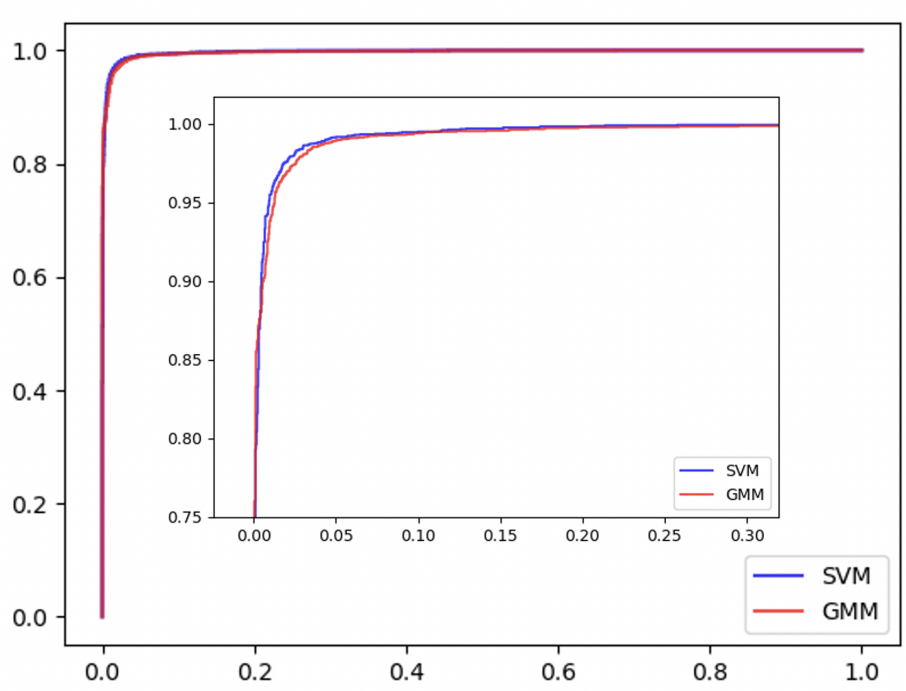


Figure 12: The ROC curve for the SVM model with RBF kernel and the GMM model with full convolution matrix and 64 components Note: The both ROC plots are the same. The inner one is zoomed version of outer one.

The ROC curve 12 shows and confirms us the both models perform similar in unbalanced applications. However, the SVM model performs better than the GMM model in the balanced application.

Until here, we always considered minimum detection cost. Now we will experiment actual costs that we would pay in real case scenario. Therefore, we will focus on actual detection cost measurement.

Table 8: Actual Determination Costs

	$\pi = 0.5$		$\pi = 0.1$		$\pi = 0.9$	
	minDCF	actDCF	minDCF	actDCF	minDCF	actDCF
SVM	0.042	0.044	0.121	0.886	0.124	0.9
GMM	0.049	0.051	0.148	0.151	0.139	0.144

As it can be observable in table 8, the actual costs of the GMM model are pretty close to the minimum costs. However, it cannot be said for the

SVM model. Since non-probabilistic scores may not provide calibrated scores, it is exactly as expected. And the SVM model cannot be delivered before optimization by calibration or an efficient threshold.

Additionally for the GMM model, the training and the validation sets should be similar to produce this kind of calibrated likelihood ratios.

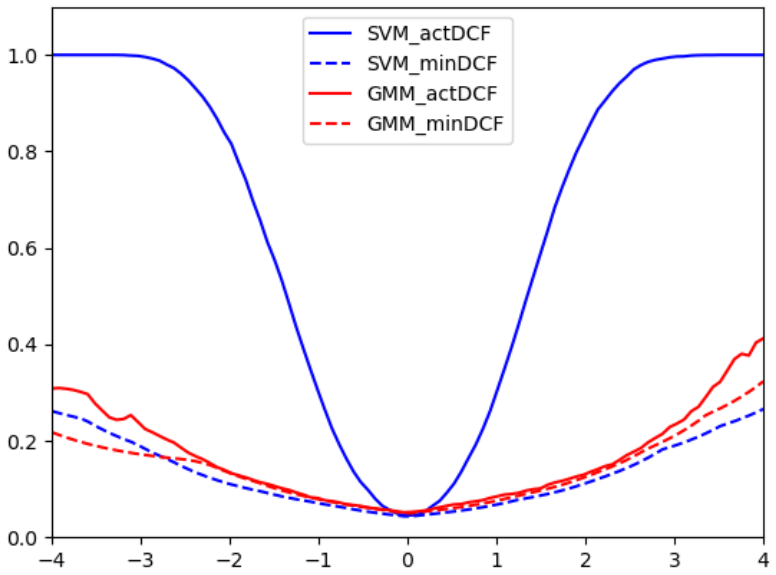


Figure 13: Bayes error plot on validation set

We can see more clearly the detection costs on particular applications on the Figure 13. And the SVM scores are uncalibrated especially for unbalanced applications. The differences between minimum and actual cost is dramatic.

The methodology of finding efficient threshold is;

1)The scores are shuffled and separated into two subset.

2)One subset is used for the optimum threshold then applied to the other.

3)For the optimum threshold selection, the threshold that performs the minimum detection cost is selected.

For the table 9, the outcome is as expected all the actual costs are gotten better dramatically for all applications in the SVM model. However, it cannot be said for the GMM model since it was already well fit. Using GMM with the theoretical threshold would be better than the efficient threshold. To see the details of the procedure we can look at the following Bayes error plot 14.

For the figure 14, it can be said that the calibration made the SVM model much better than before. But we cannot say the same argument for the GMM

Table 9: Actual Detection Costs

	minDCF	actDCF t=-log($\pi/1-\pi$)	actDCF t^*
		$\pi = 0.5$	
SVM	0.042	0.046	0.046
GMM	0.046	0.05	0.051
		$\pi = 0.1$	
SVM	0.129	0.886	0.141
GMM	0.156	0.168	0.168
		$\pi = 0.9$	
SVM	0.096	0.901	0.131
GMM	0.118	0.127	0.141

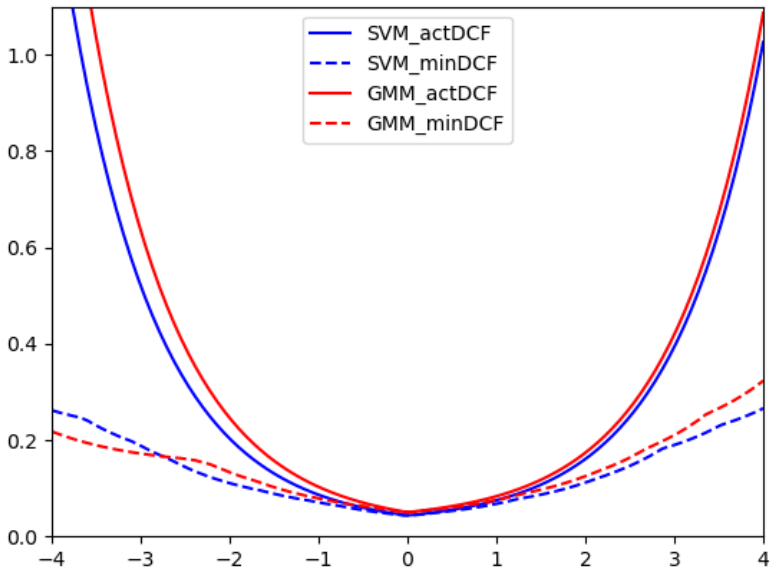


Figure 14: Bayes error plot on validation set with efficient threshold.

model too.

Overall, an effective threshold would use for balanced applications but it cannot give a general solution.

Additionally, a prior-weighted Logistic Regression model can be used to calibrate the scores of the best models. Basically, the scores and the labels will be sent as input to the LR model. Then we will recover the scores as in the following equation 6.

$$f(s) = \alpha s + \beta = \alpha s + \beta' - \log \frac{\pi}{(1-\pi)} \tag{6}$$

Where s is scores, α and β are learned parameter by the LR model. The same split used as before. And for actual costs, the theoretical threshold which equals $-\log \frac{\pi}{(1-\pi)}$ is used.

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
minDCF			
SVM	0.042	0.129	0.096
GMM	0.046	0.156	0.118
actDCF - uncalibrated			
SVM	0.046	0.886	0.901
GMM	0.05	0.168	0.127
actDCF - SVM			
Log-Reg ($\pi = 0.5$)	0.045	0.142	0.11
Log-Reg ($\pi = 0.1$)	0.062	0.164	0.185
Log-Reg ($\pi = 0.9$)	0.048	0.156	0.111
actDCF - GMM			
Log-Reg ($\pi = 0.5$)	0.048	0.163	0.145
Log-Reg ($\pi = 0.1$)	0.091	0.243	0.252
Log-Reg ($\pi = 0.9$)	0.09	0.225	0.243

Table 10: Actual and minimum detection costs of calibrated and uncalibrated models.

For the table 10, different priors for the logistic regression do not improve our result. This is understandable. Because as it has already been mentioned many times, the data is balanced. And further touches would not affect in a good way.

However, the calibration with prior equals 0.5 improved the results significantly. It can be seen clearly in the next figure 15.

As it can be observable in figure 15, the Logistic Regression approach performed better. And it gives an accurate and more general solution.

Furthermore, a score-level fusion will be applied to the best two models to improve the results. The final model will be achieved by this fusion. The scores will be separated into two subsets. Then one of them will give logistic regression as input. Then the output parameters will be used on the other subset of scores to obtain calibrated and fusion-ed new scores. And the equation as following,

$$f(s_{gmm}, s_{svm}) = \alpha s_{gmm} + \alpha s_{svm} + \beta_{gmm} + \beta_{svm} \tag{7}$$

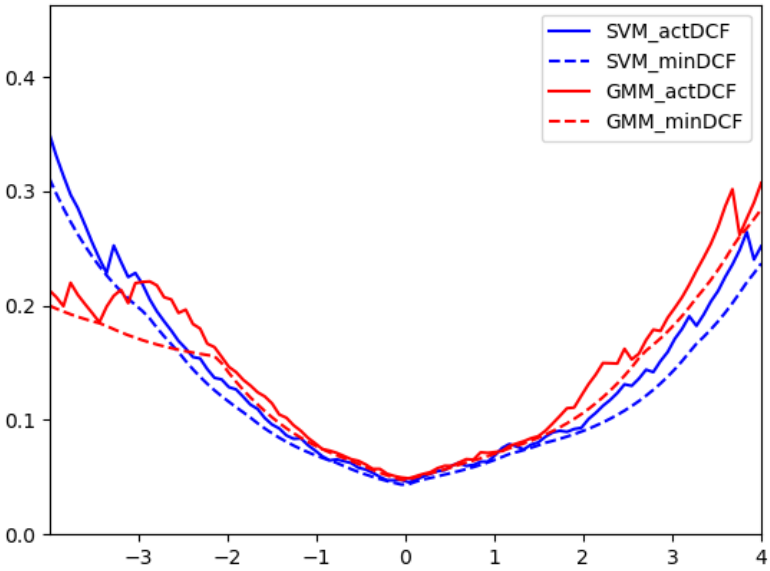


Figure 15: Bayes error plot on validation set with calibrated scores

Where s_{gmm} and s_{svm} are indicated scores of models, α and β are learned parameters by the prior-weighted LR model. Since the best results are achieved by the prior that equals 0.5, $-\log\frac{\pi}{(1-\pi)}$ was not indicated in the equation. Because it equals 0.

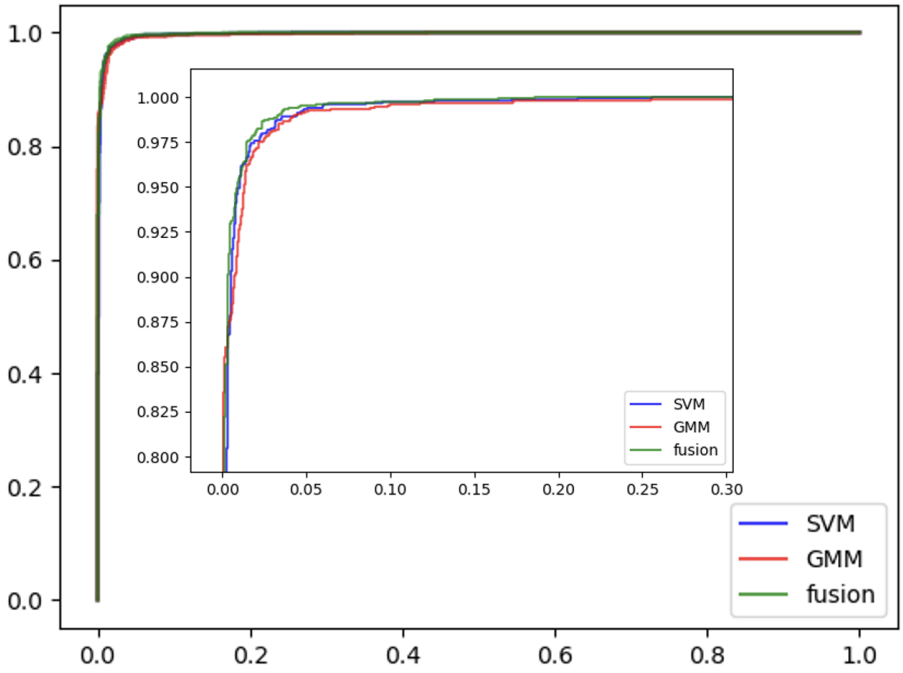


Figure 16: The ROC curve for the calibrated SVM model with RBF kernel, the calibrated GMM model with full convolution matrix and 64 components and their fusion model. Note: The both ROC plots are the same. The inner one is zoomed version of outer one.

As we can see in the figure 16, the fusion model performs better than the best two models.

Table 11: Actual Detection Costs

	$\pi = 0.5$		$\pi = 0.1$		$\pi = 0.9$	
	minDCF	actDCF	minDCF	actDCF	minDCF	actDCF
SVM	0.042	0.045	0.129	0.144	0.096	0.111
GMM	0.046	0.048	0.156	0.163	0.118	0.145
Fusion	0.037	0.039	0.111	0.147	0.089	0.108

As it can be seen in the table 11, the fusion model performed the best except for the application that has prior = 0.1. However, the difference is low for this application.

Additionally, fusion model is effective enough to reduce the minimum cost for all the applications.

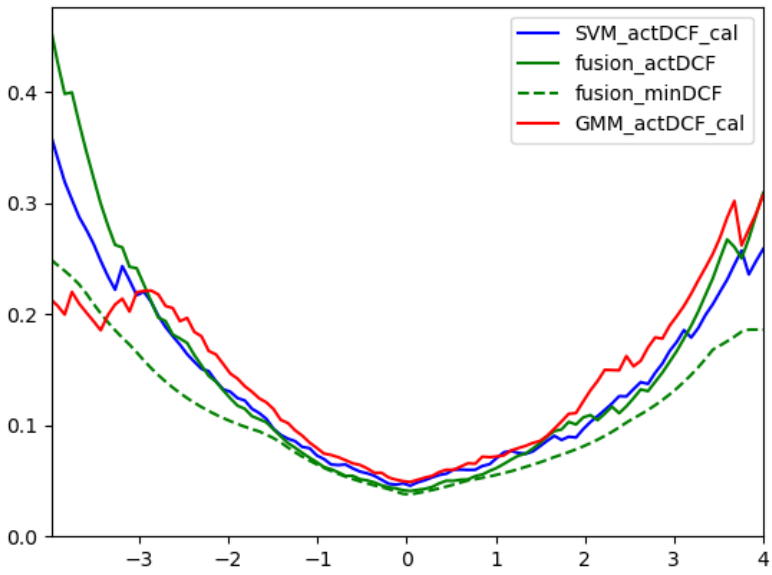


Figure 17: Bayes error plot on validation set with calibrated scores and fusion model

For the figure 17, the fusion model had the minimum costs, especially for balanced applications. But it may be said that the fusion model has some calibration issues for unbalanced applications.

Overall, the final model is the fusion model that consists of an SVM model that has an RBF kernel and a GMM model that has 64 components with a full covariance matrix. Both models are trained on raw features.

3.2 Evaluation

The evaluation part will show us the real performance of the selected model. If the test and training sets do not have differences, the same minimum detection costs can be reachable.

The same methodology will be kept in the evaluation set. %100 of the training data will be used in training.

It will be able to observe whether our selected best models could give the same results also in the evaluation set or not. The evaluation part is only for analysis of the selected model and the selection methodology. There will not

be any changes in the next parts even if our findings are different from the validation part. Because of that, the selected models in the tables will be the same as in validation.

3.2.1 Multivariate Gaussian

Table 12: MMVG - Minimum Detection cost on Evaluation Set

Single Fold						
	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Model	**Gaussianized**			**Raw Data**		
Full-Cov	0.073	0.201	0.182	0.053	0.134	0.137
Naive-Cov	0.546	0.791	0.845	0.57	0.809	0.881
Tied-Cov	0.069	0.184	0.177	0.05	0.132	0.135
Naive-Tied-Cov	0.569	0.806	0.849	0.592	0.815	0.884
PCA-11						
Full-Cov	0.049	0.133	0.133	0.052	0.136	0.139
Naive-Cov	0.064	0.174	0.156	0.069	0.191	0.175
Tied-Cov	0.049	0.129	0.132	0.051	0.136	0.134
Naive-Tied-Cov	0.057	0.131	0.144	0.051	0.136	0.134
PCA-10						
Full-Cov	0.051	0.135	0.131	0.052	0.136	0.139
Naive-Cov	0.064	0.177	0.159	0.067	0.198	0.18
Tied-Cov	0.051	0.135	0.133	0.053	0.138	0.134
Naive-Tied-Cov	0.061	0.131	0.155	0.053	0.138	0.134
LDA-1						
Full-Cov	0.051	0.133	0.135	0.05	0.132	0.135

For the table 12, the result is surprising for the MVG model that has tied covariance matrix with PCA-11 on the gaussianized data. This model performed better than all the other models for all types of applications. We will see that, will these findings change the final decision?

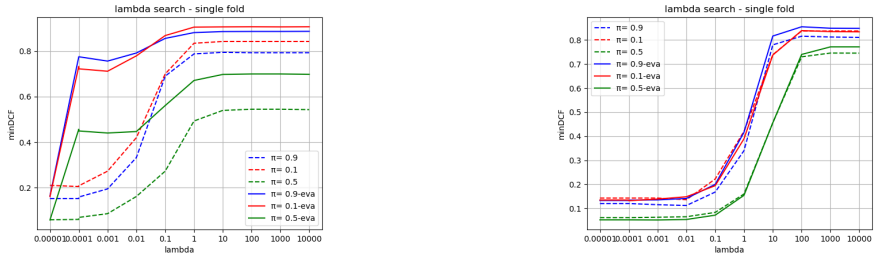
Additionally, general findings are as expected. The costs are lower on raw features than gaussianized features generally. The validation bests and the current best have minimum costs that are not too different. And it can be observable in next table 13. However, the previous decisions could not be confirmed.

Table 13: MVG - Minimum Detection cost on Evaluation Set - The Bests

Single Fold			
	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Model	**Raw**		
PCA-9			
MVG (Full-Cov)	0.053	0.144	0.14
MVG (Tied-Cov)	0.054	0.142	0.135
LDA-1			
MVG (Full-Cov)	0.05	0.132	0.135
Gaussianized & PCA-11			
Tied-Cov	0.049	0.129	0.132

3.2.2 Linear Logistic Regression

As we can see on the figure 18, there are some minor changes with comparison to validation version. However, this changes does not change our decision contrarily support our previous decision is still the best.



(a) Gaussianized data

(b) Raw data

Figure 18: Lambda search with single fold on evaluation data

For the figure 18, the minimum costs with different lambdas are very similar in comparison with validation and evaluation data, especially for raw features. And since all the costs are lower on raw data than gaussianized data. The previous lambda selection is confirmed.

Note: The minimum cost for the LR model has lambda equals 10 on gaussianized data is 0.059 for the balanced application. And this is higher than the best LR model on raw data as we can observe in the table 14.

Table 14: LR - Minimum Detection cost on Evaluation Set

Single Fold			
	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Model		**Raw**	
Full Feature			
LR ($\lambda = 10^{-4}$)	0.052	0.135	0.133
PCA-11			
LR ($\lambda = 10^{-4}$)	0.053	0.132	0.133
PCA-8			
LR ($\lambda = 10^{-4}$)	0.052	0.144	0.135
PCA-9			
MVG (Full-Cov)	0.053	0.144	0.14
MVG (Tied-Cov)	0.054	0.142	0.135
LDA-1			
MVG (Full-Cov)	0.05	0.132	0.135
Gaussianized & PCA-11			
Tied-Cov	0.049	0.129	0.132

For the table 14, the findings of logistic regression models are as expected. If we ignore the new best model, the other results are the same as the validation results.

3.2.3 Linear Support Vector Machine

The hyperparameter selection by validation data is confirmed as we can see in the figure 19b. The best results are obtained with the same hyperparametes which are $K = 10^{-1}$ and $C = 10$. (The most blue bar is better.)

Note: Since we did not consider gaussianized versions of SVM models on validation data, also they will now be considered in evaluation data. But the results are worse than raw data.

(a) All search space

(b) Narrowed search space

Figure 19: Hyper-parameter tuning (C and K) on evaluation set with single fold.

Table 15: Linear SVM - Minimum Detection cost on Validation Set

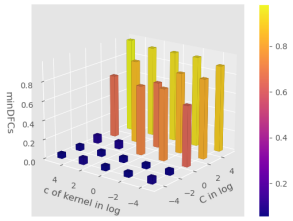
Single Fold			
	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Model		**Raw**	
Full Feature			
SVM ($C = 10$ & $K = 10^{-1}$)	0.05	0.133	0.132
SVM ($C = 10$, $K = 10^{-1}$, $\pi = 0.5$)	0	0	0
PCA-11			
LR ($\lambda = 10^{-4}$)	0.053	0.132	0.133
SVM ($C = 10$ & $K = 10^{-1}$)	0.051	0.135	0.137
PCA-10			
SVM ($C = 10$ & $K = 10^{-1}$)	0.268	0.505	0.534
PCA-8			
LR ($\lambda = 10^{-4}$)	0.052	0.144	0.135
PCA-9			
MVG (Full-Cov)	0.053	0.134	0.137
MVG (Tied-Cov)	0.05	0.132	0.135
LDA-1			
MVG (Full-Cov)	0.05	0.132	0.135
Gaussianized & PCA-11			
Tied-Cov	0.049	0.129	0.132

As it can be observable in table 15, the best model of SVM that with full features, performed close to the MVG model that has a full covariance matrix with PCA-11 on gaussianized data (the current best model). If we still ignore

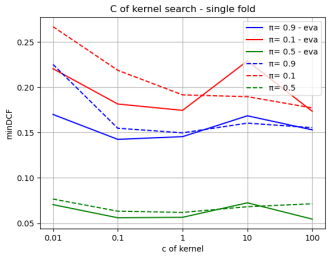
the current best model, we can say those results are confirmed by our previous decisions.

3.2.4 Quadratic Support Vector Machine with Kernel

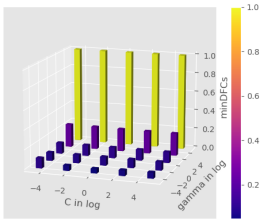
For the figure 20, the hyperparameter selections of both quadratic SVM models with polynomial and RBF kernels are exactly the same in comparison with validation data. It can be said that our decisions for the hyperparameters are confirmed. The selections again, $C = 1$, $\gamma = 10^{-2}$ for the SVM model with a RBF kernel and $C = 10^{-2}$, constant of kernel=1 for the SVM model with a polynomial kernel.



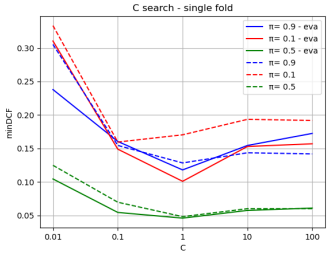
(a) particular constant and C combinations for the polynomial kernel



(b) Clear search of constant of kernel with $C = 10^{-2}$



(c) particular gamma and C combinations for the RBF kernel



(d) Clear search of C with $\gamma = 10^{-2}$

Figure 20: Hyperparameter tuning for the quadratic SVM classifiers with kernel on raw data. The top graphs are for the polynomial kernel, bottom graphs are for RBF kernel

As it can be observable in the table 16, the SVM model with RBF model performed significantly better than all the other models, especially for the unbalanced applications.

Additionally, since the number of samples are exactly the same for the both classes, re-balancing with $\pi = 0.5$ changes nothing.

Overall, the previous decision for the SVM model which was one of the best

Table 16: Kernel SVM - Minimum Detection Cost on Evaluation Set

Single Fold			
	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Model	**Raw**		
Full Feature			
Poly Kernel SVM ($C = 10^{-2}, c = 1$)	0.056	0.174	0.145
RBF Kernel SVM ($C = 1, \log \gamma = -2$)	0.046	0.101	0.118
RBF Kernel SVM ($\pi = 0.5$)	0.046	0.101	0.118
RBF Kernel SVM ($\pi = 0.1$)	0.049	0.13	0.147
RBF Kernel SVM ($\pi = 0.9$)	0.048	0.132	0.125
Linear SVM ($C = 10$ & $K = 10^{-1}$)	0.05	0.133	0.132
PCA-10			
Linear SVM ($C = 10$ & $K = 10^{-1}$)	0.268	0.505	0.534
PCA-8			
LR ($\lambda = 10^{-4}$)	0.052	0.144	0.135
Gaussianized & PCA-11			
Tied-Cov	0.049	0.129	0.132

models is confirmed.

3.2.5 Gaussian Mixture Model

For the figure 21, the results for the GMM models are as expected, except the model with full covariance matrix. Actually, the over-fitting was in our expectation but for the higher number of components.

As we can observe in graph 21a, there is an overfitting problem after the number of components equals 32. It can be said even now, our final decision is not confirmed by this results.

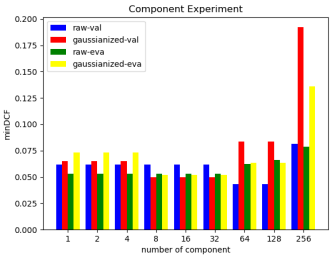
Table 17: GMM - Minimum Detection Cost on Evaluation Set

Single Fold			
	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
Model	**Gaussianized**		
RBF Kernel SVM ($C = 1, \log \gamma = -2$)	0.046	0.101	0.118
RBF Kernel SVM ($\pi = 0.1$)	0.049	0.13	0.147
RBF Kernel SVM ($\pi = 0.9$)	0.048	0.132	0.125
GMM (Full Cov / 64-Raw)	0.062	0.153	0.186
GMM (Naive Cov / 256-Raw)	0.088	0.221	0.239
GMM (Tied Cov / 1024-Raw)	0.053	0.134	0.137
GMM (Naive Tied Cov / 64-Raw)	0.087	0.235	0.208

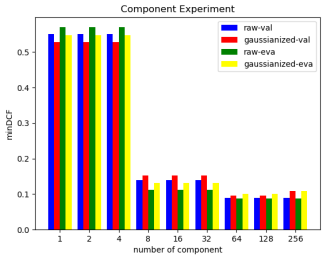
The results are totally different than we expected as it can be seen in the table 17. It can be said that the final decision decision which is the GMM model with a full covariance matrix is worse than almost all of the models including the simple models such as MVG models. The final decision is not confirmed.

Overall, the consideration between validation and evaluation sets are;

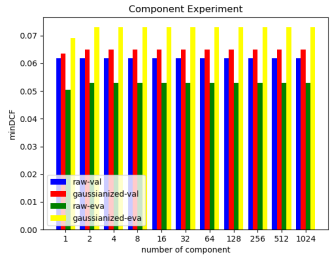
1) The GMM model selection was totally wrong about everything. It performed awful in evaluation data.



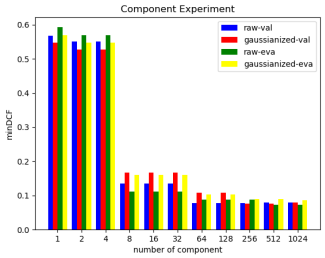
(a) Full Covarianve Matrix



(b) Naive Covarianve Matrix



(c) Tied Covarianve Matrix



(d) Naive Tied Covarianve Matrix

Figure 21: Number of component search on evaluation data.

2) The SVM model selections were totally right for both the hyperparameter selection and the kernel selection.

3.2.6 Analysis

For the figure 22, the fusion mode was performed better even if the GMM model selection was wrong.

As we can see in the table 18, the minimum detection costs are slightly higher than the SVM model in unbalanced applications. However, the fusion model has the lowest minimum detection cost with a significant difference for the balanced application.

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
SVM	0.046	0.101	0.118
GMM	0.062	0.153	0.186
Fusion	0.041	0.102	0.119

Table 18: Minimum detection cost on evaluation data for fused model

Again until here, we always considered minimum detection cost for the models on evaluation data. Now, we will see actual costs for the models.

Two different optimization techniques were discussed in the validation section. The first one was the efficient threshold which the threshold that is used

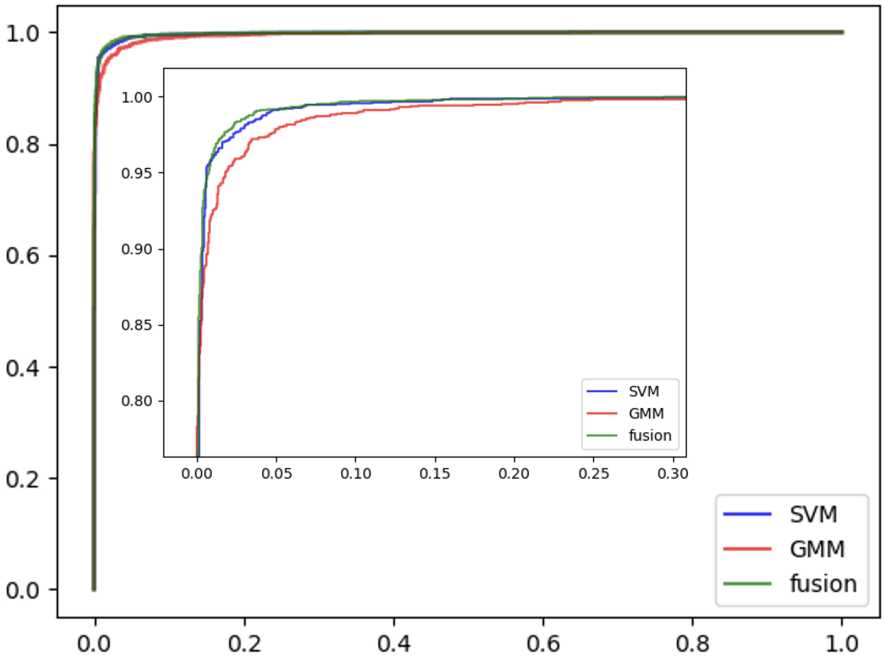


Figure 22: The ROC curve for the SVM model with RBF kernel, the GMM model with full convolution matrix and 64 components and the fusion model. Note: The both ROC plots are the same. The inner one is zoomed version of outer one.

for minimum detection cost is picked. The other one is calibration with the prior-weighted Logistic Regression model. The calibration method gave a more accurate and more general solution whole efficient threshold was good for the balanced application but it cannot be said for unbalanced applications.

The efficient threshold and the calibration parameters were found from validation data scores.

	$\pi = 0.5$	$\pi = 0.1$	$\pi = 0.9$
minDCF			
SVM	0.046	0.101	0.118
GMM	0.062	0.153	0.186
actDCF - Efficient Threshold			
SVM	0.05	0.107	0.118
GMM	0.066	0.159	0.190
actDCF - Calibration			
SVM	0.049	0.118	0.132
GMM	0.064	0.158	0.198

Table 19: Actual and minimum detection costs of models on evaluation data.

We can observe in the table 19, surprisingly the efficient threshold is better for optimization of evaluation scores.

Table 20: Actual and minimum detection costs for the fusion model

	$\pi = 0.5$		$\pi = 0.1$		$\pi = 0.9$	
	minDCF	actDCF	minDCF	actDCF	minDCF	actDCF
SVM	0.046	0.049	0.101	0.118	0.118	0.132
GMM	0.062	0.064	0.153	0.158	0.186	0.198
Fusion	0.041	0.041	0.102	0.119	0.119	0.123

For the table 20, the fusion performed better than the other models in balanced and unbalanced applications with prior equals 0.9. However, the potential (minimum cost) of the fusion model is slightly worse than the SVM model for unbalanced applications. Moreover, the fusion better for both minimum and actual costs for balanced applications.

As it can be seen in the figure 23, the fusion model performs best for balanced applications. However, it cannot be said for the unbalanced applications.

The reason that the fusion model does not perform best significantly is the selection of the second model which is the GMM model. Maybe the GMM model that was selected performs worse than many models including simple models such as MVG models. And this situation affects the performance of the fusion model significantly.

4 Conclusions

In a conclusion, because of the weak validation technique single fold. One of the final decisions was wrong. And this wrong selection affected the final fusion model performance. However, even though one of the best model selection

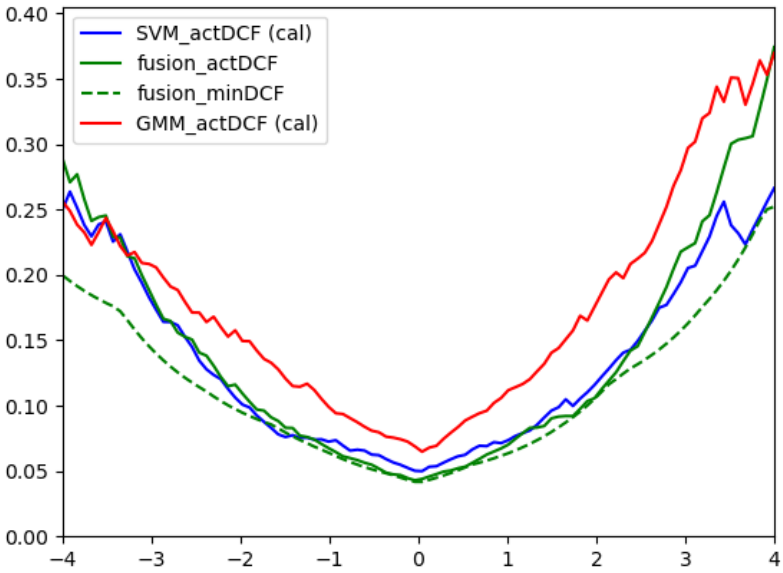


Figure 23: Bayes error plot on evaluation test.

models was wrong, the fusion model was the best. But the fusion model performance could be increased by a different selection. Therefore, the importance of validation was exhibited in this report. All the model selection should have been done by rich validation techniques such as k-fold cross-validation.

In detail, we observed over-fitting in the GMM models with full covariance matrices. However, it was happening on the model that has 256 components on raw data. Because of that, even if the models with 64 and 128 components performed the same. The GMM model with 64 components has been selected due to over-fitting observation. But even this action could not make the final model avoid this issue.

The decisions that are made by validation data, were partially ineffective for evaluation data for the model selection.

0.041 detection cost were reached for primary application with the fusion model.