INTRODUCTION

The goal of the project is the development of a classifier able to verify the correctness of an identity claim based on biometric characteristics of a subject, in details through spoken utterance. The task is to compare an unknown utterance with those belonging to the claimed identity and check if they are similar or not.

The problem can be cast as a binary problem over pairs of utterances. Given a pair of utterances the classifier should decide if the unknown one and the claimed identity one correspond to the same speaker (label 1) or different speaker (label 1).

In this use case the utterances are represented in terms of speaker embeddings. The embeddings are 5-dimensional, continuous-valued vectors, and features have no specific physical interpretation.

INFORMATION ABOUT THE DATASET

Each sample of the dataset is composed of a pair of embeddings for a total dimension of 10.

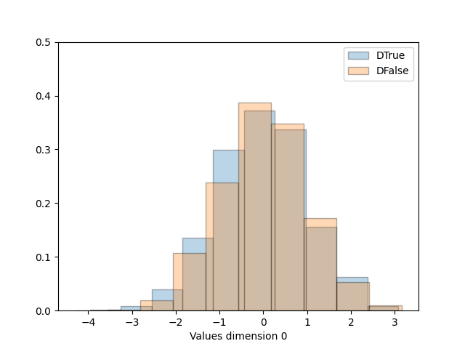
The training dataset is composed of 3500 samples: 43% belong to label 1 and 57% to class 0.

The test dataset instead is composed of 7500 samples in total: 33% belong to label 1 and 67% to class 0.

Both datasets are slightly imbalanced, with the different speaker class (label 0) having slightly more samples. Pairs belong to either the male or female gender, which is not provided by the datasets.

DATASET FEATURES ANALYSIS

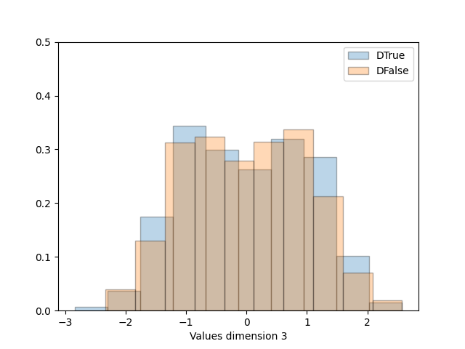
For the analysis of the dataset (which has been normalized first using z-score) is crucial to plot the distribution of the features over histograms. For each feature has been made a distinction between samples belonging to different classes.

A graph of values

Description automatically generatedA graph of values

Description automatically generated

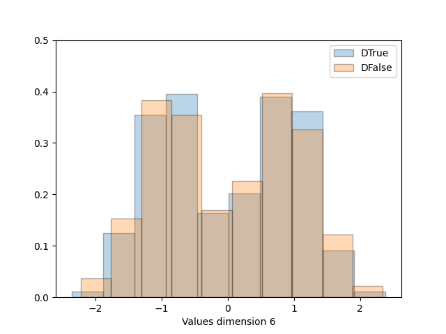
Figure Figure Figure

A graph of values

Description automatically generatedA graph of values

Description automatically generated

Figure Figure Figure

A graph of values

Description automatically generatedA graph of values

Description automatically generated

Figure Figure Figure

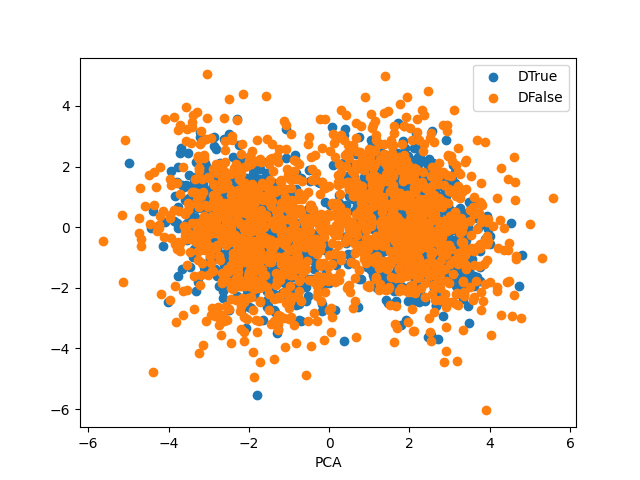
A graph of values

Description automatically generated

Figure

All feature distributions follow a gaussian one with an exception for feature 1, 3, 6, 8. These last reflect a gaussian with 2 components due to the presence of both genders, male and female.

To have a general view over our dataset it has been applied a PCA transformation with 2 dimensions on the left and a LDA transformation on the right. In the PCA graph is evident the presence of two clusters for both the class and in the LDA graph the distribution of the single feature of both classes overlap. These two clusters can be explained by the presence of spoken utterance belonging to the male and female gender. With these results is expected that the data cannot be well separated linearly so a non-linear model should perform better in the classification task.

A graph of a number of objects

Description automatically generated with medium confidence

Figure Figure

DATASET FEATURES CORRELATION ANALYSIS

The graphs below represent the correlation between each feature of the training dataset.

Each Heat Map shows the absolute value of the Pearson Correlation Coefficient. On the left it has been performed on the whole dataset and on the right the graph of the Explained Variance.

A graph of a crossword puzzle

Description automatically generatedA graph with a line

Description automatically generated

Figure Figure

Looking at figure 13 features 2-7 and 4-9 are the highest correlated. For what concern the others one there is a slight correlation such as also no correlation between some of them. Figure 14 tells us that removing two dimensions will keep very high the variance of the dataset so without losing a lot of information which is crucial during the training phase of our classifier.

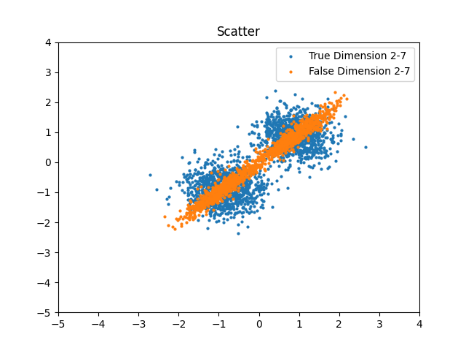
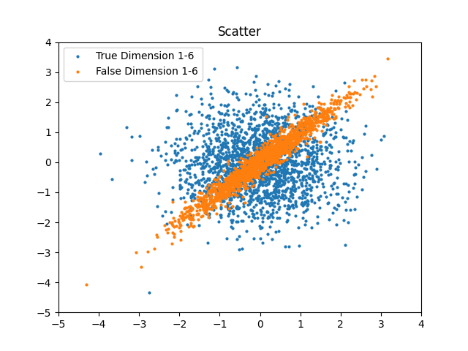
Below there are the heatmap of the Pearson correlation coefficient of the dataset features belonging respectfully on the left to the label 0 (False) and on the right to the label 1 (True). They are pretty similar with exception for features (1-6), (2-7), (3-8), (4-9), (5-10) which are more correlated in the False samples compared to the True samples. The higher correlation can be seen clearly in the scatter plot immediately below the heatmaps.

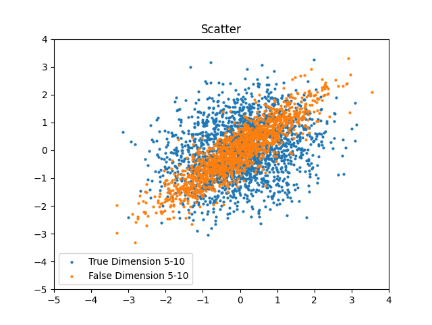
A close-up of a graph

Description automatically generatedA blue squares with white lines

Description automatically generated with medium confidence

A graph of scatters with orange and blue dots

Description automatically generatedA graph of scatters

Description automatically generated

With this data we can deduce that an MVG classifier should perform better.

QUI VA AGGIUNTA LA PARTE DEL NAÏVE BAYES E DEL TIED COVARIANCE. VA FATTA L’ANALISI DEI DUE MODELLI ANALIZZANDO LE DUE HEATMAP DI CUI SOPRA

TRAINING PROTOCOL

For the training phase it has been used the K-fold cross validation with 5 folds.

The target application working point given by the project is defined by the triple (π̃= 0.1, Cf n = 1, Cf p = 1). During this phase two others application working points have been used: (π̃ = 0.5, Cf n = 1, Cf p = 1) and (π̃ = 0.9, Cf n = 1, Cf p = 1).

The performance of the different classifiers is measured in terms of Detection Cost Function (DCF). The classifier with the best performance will be the one with the lowest DCF.

GAUSSIAN CLASSIFIER

In this section the classifier has been trained with several gaussian models: MVG (Multi Variate Gaussian), NB (Naïve Bayes), TC (Tied Covariance) and TNB (Tied Naïve Bayes). The first table shows the performance of the dataset in terms of minimum DCF without pre-processing. The following ones show the performances when a pre-process technique has been applied. If applied the type of pre-process technique is described at the top of the table.

RAW Z-Score

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| **Model** | πT=0.1 | πT=0.5 | πT=0.9 | πT=0.1 | πT=0.5 | πT=0.9 |  |
|  |  |  |  |  |  |  |  |
| MVG | 0.255 | 0.078 | 0.140 | 0.253 | 0.078 | 0.140 |  |
| NB | 0.998 | 0.960 | 0.996 | 0.999 | 0.960 | 0.996 |  |
| TC | 1.000 | 0.937 | 0.990 | 0.999 | 0.937 | 0.990 |  |
| TNB | 0.996 | 0.963 | 0.999 | 0.998 | 0.963 | 0.999 |  |
|  |  |  |  |  |  |  |  |

As expected from the previous consideration in the Dataset Feature Analysis, since the Tied Covariance model uses a linear separation surface for the classification task it performs badly. The same is for the Tied Naïve Bayes model.

Naïve Bayes applied on the dataset without pre-processing performs bad. This model assumes the independence between the features by diagonalizing the covariance matrix, but we have seen from the heatmaps a strong correlation among some attributes especially in the samples belonging to the False label (label 0).

MVG is the one performing better among all because it uses a quadratic separation rule and the features’ distribution fit well into a gaussian.

RAW PCA 8 Z-Score PCA 8

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| **Model** | πT=0.1 | πT=0.5 | πT=0.9 | πT=0.1 | πT=0.5 | πT=0.9 |  |
|  |  |  |  |  |  |  |  |
| MVG | 0.256 | 0.078 | 0.148 | 0.308 | 0.099 | 0.184 |  |
| NB | 0.998 | 0.946 | 0.992 | 1.000 | 0.946 | 0.992 |  |
| TC | 1.000 | 0.919 | 0.981 | 1.000 | 0.919 | 0.981 |  |
| TNB | 0.990 | 0.953 | 0.998 | 0.992 | 0.953 | 0.998 |  |
|  |  |  |  |  |  |  |  |

RAW PCA 9 Z-Score PCA 9

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| **Model** | πT=0.1 | πT=0.5 | πT=0.9 | πT=0.1 | πT=0.5 | πT=0.9 |  |
|  |  |  |  |  |  |  |  |
| MVG | 0.257 | 0.080 | 0.141 | 0.308 | 0.098 | 0.170 |  |
| NB | 0.998 | 0.946 | 0.992 | 1.000 | 0.946 | 0.992 |  |
| TC | 1.000 | 0.919 | 0.981 | 1.000 | 0.919 | 0.981 |  |
| TNB | 0.990 | 0.953 | 0.998 | 0.992 | 0.953 | 0.998 |  |
|  |  |  |  |  |  |  |  |

We have applied PCA as pre-processing method on all the gaussian classifiers. We performed the classification task reducing by a maximum of 2 the number of the database-s dimension and it ends up that working with 8 directions has the best result.

The MVG classifier keeps its good performances.

Naïve Bayes has a significant improvement. One of the effects of PCA is to decorrelate features, so the model assumption (independence among the features) holds well.

Tied Covariance and Naïve Tied Covariance continue to have bad performances for the same reason explained above.

In all the results obtained the Z-Score normalization gives equal and slightly worst results.

DISCRIMINATIVE CLASSIFIER

In this part the dataset has been trained with discriminative classifiers. Two types of models have been implemented: the Logistic Regression model (LR) and the Support Vector Machine one (SVM). Along with them we have used also the Quadratic Logistic Regression (QLR) and the kernel Support Vector Machine applying the Polynomial and Radial Basis kernel function. The last two use a quadratic separation surface as classification rule.

For what concern LR its goal is to find the best hyperplane that maximizes the posterior probability. This problem can be casted to the minimization of the objective function which has as input an hyperparameter called Regularization Term (λ). The right value of λ has been found using cross-validation techniques using a range which goes from 10-5 to 105.

----------------------------------------------------------

The goal of Lr is to find the best hyperplane that maximizes the posterior probability(w^tx+b),to make the problem solvable we will introduce a regularization term, otherwise the best solution will be anyways with w-> inf .

The Regularization term will be found using cross-validation techniques from 10-5 to 105

----------------------------------------------------------

During the training phase the log-likelihood ratio has been performed using the class prior probabilities in the list: [(πT = 0.1, πF = 0.9), (πT = 0.5, πF = 0.5), (πT = 0.9, πF = 0.1)]. For each of them the Detection Cost Function has been performed using all the application working points listed in the “TRAINING PROTOCOL” section.

In the figure below we have plotted the value of DCFs as the Regularization term changes.

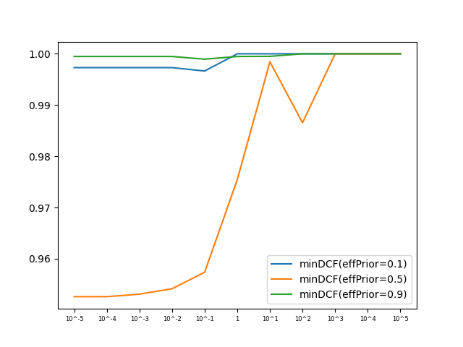
A graph with numbers and lines

Description automatically generatedA graph with green and orange lines

Description automatically generatedA graph of a number of numbers

Description automatically generated with medium confidence

Figure Zscore Figure Zscore Figure Zscore

A graph of a number of numbers and a line

Description automatically generated with medium confidenceA graph of a number of numbers

Description automatically generated with medium confidence

Figure Raw Figure Raw Figure Raw

The Raw dataset shows slightly better performance compared to the Zscore normalized one. In most the case the best value of DCF is given choosing the value 10-5 for the hyper parameter λ.

As we can see from the plots the results from raw Dataset and z-score are very similar, and not really interesting because as we said in the past, linear models does not fit well for our data.

The best value for hyper parameter λ , looking the graph, is definitely 10^-5 in most cases.

In order to analyze the performance , we will consider the following table:

The table below shows the performance of the LR classifiers given the value chosen for the Regularization Term

RAW Z-Score

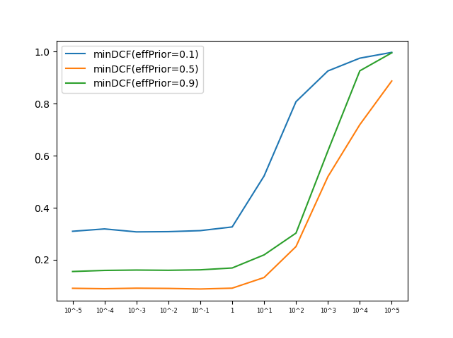
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | λ=10-5 |  |  | λ=10-5 |  |  |
| **Model** | π̃=0.1 | π̃ =0.5 | π̃ =0.9 | π̃ =0.1 | π̃ =0.5 | π̃ =0.9 |  |
|  |  |  |  |  |  |  |  |
| LR (πT =0.1) | 0.996 | 0.952 | 0.998 | 0.997 | 0.952 | 0.997 |  |
| LR (πT =0.5) | 1.0 | 0.915 | 0.984 | 1.0 | 0.916 | 0.984 |  |
| LR (πT =0.9) | 1.0 | 0.736 | 0.839 | 1.0 | 0.736 | 0.839 |  |
|  |  |  |  |  |  |  |  |

As mentioned in the section above since LR classifiers use a linear separation surface to classify each sample its performance are poor

The table confirm the that Logistic Regression model is not useful due to the fact that uses linear decision.

Now, our attention will shift towards the QLR version of the model. Similar to what we did with the LR version, we will use the same range of values for λ. As we've done before, we will examine the graphs of the original raw data and the Z-Score normalized .

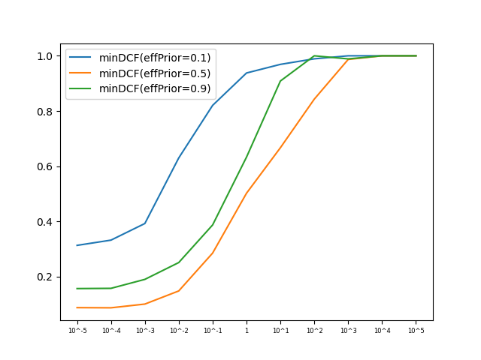
Now it’s analyzed the quadratic version of the LR classifier, the QLR. For what concern the values of the hyperparameter λ have been made the same choice did in the training phase of the LR. Below the plots of the DCFs :

A graph of different colored lines

Description automatically generatedA graph of a number of numbers

Description automatically generated with medium confidence

Figure Raw Figure Raw Figure Raw

A graph of different colored lines

Description automatically generatedA graph of different colored lines

Description automatically generated

Figure Zscore Figure Zscore Figure Zscore

As we already mentioned since this model uses a quadratic separation surface it exhibits greater performance with respect to LR. Also in this case the Raw Dataset gives out slightly better results compared to the normalized dataset. The best value of the hyper parameter λ is 10-5 in most of the cases. The chosen Regularization term’s value minimizes overfitting while still delivering comparable performance to other values. By selecting this value, we can mitigate the risk of overfitting the table shows gives in number the results of the performance,

In contrast to the Lr model, the QLR model exhibits greater performance, as we had anticipated.

This is evident from the scatter plots, histograms, and LDA analysis, all of which indicate that quadratic models can perform way better than linear ones. Also in this case z-score pre-processing does not change the results that are very similar to the raw one’s. The best value of λ, will be 10- 5 in almost all cases , so we will list dcf with 10^-5. Table->

.

RAW Z-Score

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | λ = 10-5 |  |  | λ = 10-5 |  |  |  |
| **Model** | π̃ =0.1 | π̃ =0.5 | π̃ =0.9 | π̃ =0.1 | π̃ =0.5 | π̃ =0.9 |  |
|  |  |  |  |  |  |  |  |
| QLR (πT =0.1) | 0.309 | 0.090 | 0.154 | 0.313 | 0.087 | 0.156 |  |
| QLR (πT =0.5) | 0.330 | 0.091 | 0.163 | 0.337 | 0.086 | 0.145 |  |
| QLR (πT =0.9) | 0.414 | 0.102 | 0.171 | 0.342 | 0.090 | 0.147 |  |
|  |  |  |  |  |  |  |  |

The chosen lambda value minimizes overfitting while still delivering comparable performance to other small 𝜆 values. By selecEng this value, we can miEgate the risk of overfitting.

As we can see from the table the best prior πT is 0.5 , in the training class prior and scoring class prior too, because dataset is slightly unbalanced but not that unbalanced equals to prior 0.1 and 0.9. !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! !!!

No Probabilist Model (SVM)

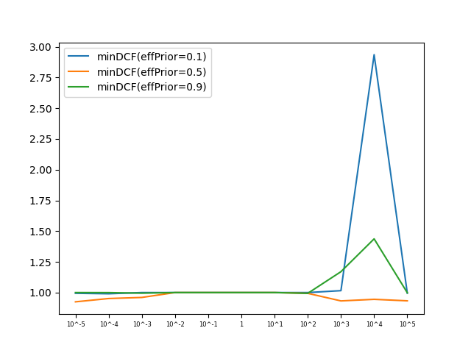
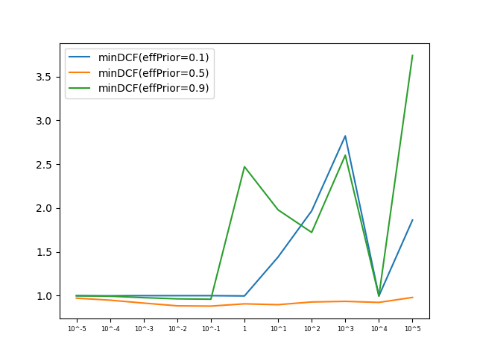
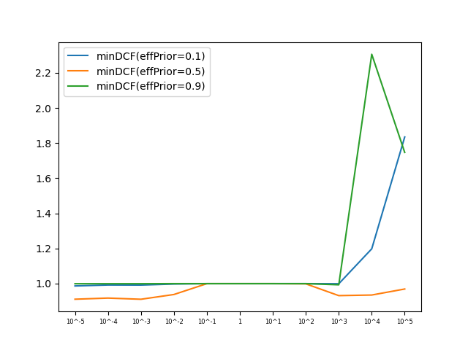
Now, we turn our attention to non-probabilistic models such as Support Vector Machine models (SVMs). They are referred to as non-probabilistic because the scores they generate do not have a probabilistic interpretation. We will begin by examining the linear SVM, and, just as we did previously, we will aim to analyze the model and its assumptions.

Similar to the LR model, the SVM model seeks to identify a hyperplane that effectively separates the classes. However, unlike the LR model, the SVM's objective is to find the hyperplane that maximizes the margin, which we refer to as the "maximum margin hyperplane." As with Discriminative models, we will consider the prior-weighted version of the model, so the objective, or the dual problem of SVM that we aim to maximize, will be:

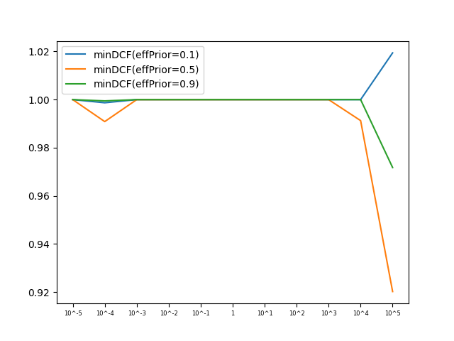
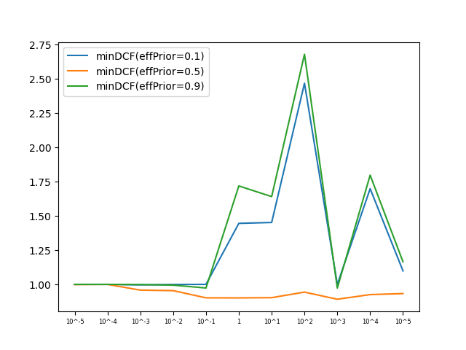
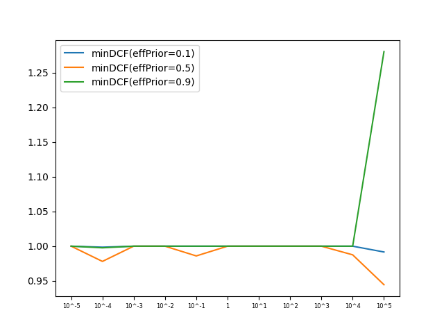
AGGIUNGI SVM TEORIA CI pesate e formule object

Once again, we have a hyperparameter to adjust, denoted as "C," and we will explore a logarithmic range from 10^-5 to 10^5. In line with our earlier findings, we generally anticipate that a quadratic decision rule will outperform a linear one when it comes to class separation.

raw



Zscore



As observed in the plots, the outcomes from the raw dataset and the z-score normalized dataset are highly comparable, albeit not particularly noteworthy. As previously mentioned, linear models are not a good fit for our data.

Upon inspecting the graphs, it becomes evident that the classifier does not provide good result ,but we will list in the table for simplicity all values coming from λ is consistently 10^-3.

RAW Z-Score

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | λ = 10-3 |  |  | λ = 10-3 |  |  |  |
| **Model** | πT=0.1 | πT=0.5 | πT=0.9 | πT=0.1 | πT=0.5 | πT=0.9 |  |
|  |  |  |  |  |  |  |  |
| SVM (πT =0.1) | 0.991 | 0.911 | 0.999 | 1.0 | 1.0 | 1.0 |  |
| SVM (πT =0.5) | 1.0 | 0.915 | 0.977 | 0.996 | 0.957 | 0.998 |  |
| SVM (πT =0.9) | 1.0 | 0.960 | 0.994 | 1.0 | 1.0 | 1.0 |  |
|  |  |  |  |  |  |  |  |

Given the data in the table we can confirm that this model performs bad.

Now we’ll move to kernel SVM , starting from Poly SVM that in our case we’ll do only the quadratic ones with kernel function as k(x,K)= (x1^-1 \*x2 +c)^2+k (where K is chosen as 1), and finally Radial Bases Svm with kernel function as k(x,gamma,K) =exp( -gamma\* ||x1-x2||^2)). Both the models depend on C and only the radial version depends on a parameter gamma that we have tuned with value 10-1,10-2,10-3.

Poly SVM

Poly SVM is a non-linear model , we will examine the graphs of the original raw data and the Z-Score normalized , and trying to choice the best C .

Raw

Immagine che contiene testo, schermata, linea, diagramma

Descrizione generata automaticamenteImmagine che contiene testo, schermata, linea, diagramma

Descrizione generata automaticamenteImmagine che contiene testo, schermata, diagramma, linea

Descrizione generata automaticamente

ZSCORE

Immagine che contiene testo, linea, schermata, diagramma

Descrizione generata automaticamenteImmagine che contiene testo, linea, schermata, diagramma

Descrizione generata automaticamenteImmagine che contiene testo, linea, schermata, diagramma

Descrizione generata automaticamente

As we can see from the graph , the Poly Svm(Quadratic SVM) has greater performance than Svm ,

the choice of C change between Raw Dataset and Z-score Dataset, while the performance are almost the same,so we are going to choice C=10^-2 for the Raw Dataset and C=10^2 for the Z-score One

As for the QLR the best prior as we can see from the table the best prior πT is 0.5 , for the same reasons .

RAW Z-Score

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | λ = 10-2 |  |  | λ = 10^2 |  |  |  |
| **Model** | πT=0.1 | πT=0.5 | πT=0.9 | πT=0.1 | πT=0.5 | πT=0.9 |  |
|  |  |  |  |  |  |  |  |
| QSVM (πT =0.1) | 0.302 | 0.087 | 0.149 | 0.329 | 0.086 | 0.148 |  |
| QSVM (πT =0.5) | 0.298 | 0.082 | 0.149 | 0.291 | 0.080 | 0.136 |  |
| QSVM (πT =0.9) | 0.423 | 0.103 | 0.173 | 0.354 | 0.090 | 0.164 |  |
|  |  |  |  |  |  |  |  |

Based on the table, we can observe that the results are really good, as for the QLR model. This reinforces our assumption that a non-linear decision rule is appropriate for this dataset. Additionally, there isn’t a notable difference in performance between the RAW and Z-Score features, with the model benefiting from the use of Z-Score normalization,as we said.

Based on the results obtained, it is evident that this model, particularly the QSVM with πT = 0.5, is among the top-performing models for classifying our sample. Therefore, we will consider this model for further analysis.

Radial Bases

Teoria…

Now we will make a decision on hyperparameters by looking at the graphs having prior πT=0.5 for Raw Dataset and on Dataset with Z-score pre-processing.

Immagine che contiene schermata, testo, diagramma, Policromia

Descrizione generata automaticamente

Immagine che contiene schermata, diagramma, testo, Policromia

Descrizione generata automaticamente Raw

We are going to chose for the Raw hyperparameters( C=10^3 and gamma 10-2 ) while for Zscore hyperparameters(C=10^1 gamma=10^-2

RAW Z-Score

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | C = 10^-3 | γ=10^-2 |  | C = 10^1 | γ=10^-2 |  |  |
| **Model** | πT=0.1 | πT=0.5 | πT=0.9 | πT=0.1 | πT=0.5 | πT=0.9 |  |
|  |  |  |  |  |  |  |  |
| RBSVM (πT =0.1) | 0.385 | 0.105 | 0.204 | 0.358 | 0.120 | 0.217 |  |
| RBSVM (πT =0.5) | 0.415 | 0.088 | 0.204 | 0.468 | 0.086 | 0.217 |  |
| RBSVM (πT =0.9) | 0.428 | 0.105 | 0.162 | 0.468 | 0.116 | 0.176 |  |
|  |  |  |  |  |  |  |  |

We can compare this model with Quadratic-SVM and the result are almost identical for Raw and Z-score .

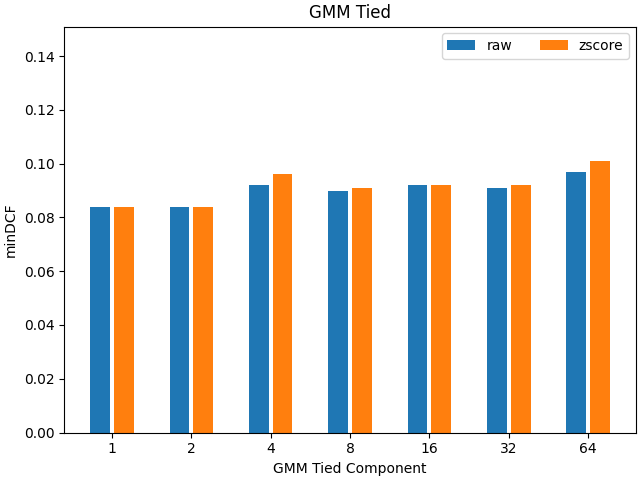
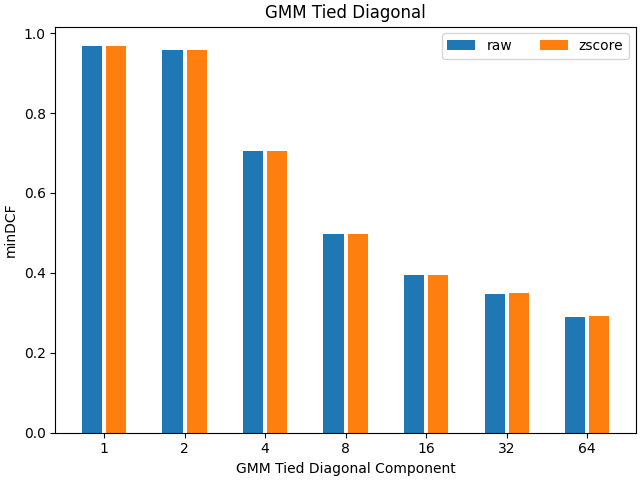
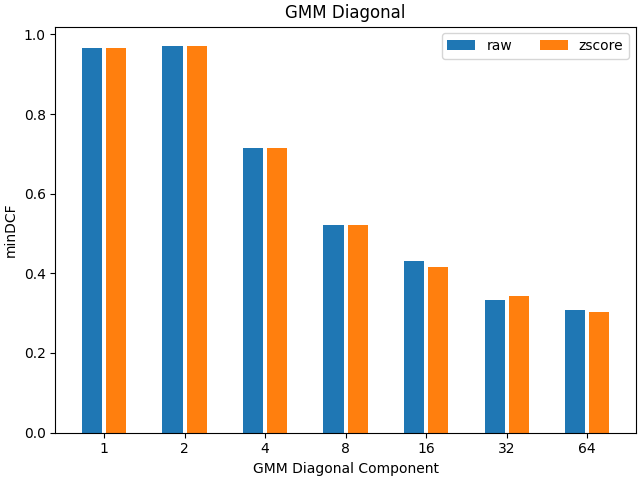
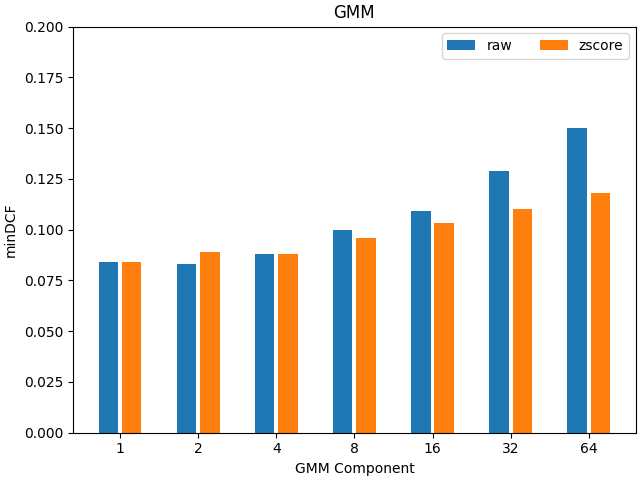
GMM

The Gaussian Mixture Model (GMM) is a type of classifier commonly used for density estimation problems. The underlying assumption when using a GMM is that the data can be approximated as a mixture of one or more Gaussian distributions. In other words, the GMM assumes that the data is generated by multiple Gaussian components, each contributing to the overall distribution of the data.

The expectation is that the best results will be achieved with Gaussian Mixture Models having 2 components. This choice is influenced by the dataset analysis, which indicates that the training data's distribution resembles a Gaussian distribution with approximately 2 components or clusters.

In this case, the hyperparameter to tune is the number of components for each Gaussian distribution within the mixture model.

Grafici



As we expected by analyze MVG results, the two diagonal models yield unsatisfactory results due to the moderate correlation observed in the heatmaps, while Tied Model in this case works well because the Tied assumption is between component of the same classes.

AGGIUNGERE TABELLE GMM E GMM PCA

The best three performing models are:

RAW Z-Score

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Model |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| RBSVM (πT =0.1) | 0.385 | 0.105 | 0.204 | 0.358 | 0.120 | 0.217 |  |
| RBSVM (πT =0.5) | 0.415 | 0.088 | 0.204 | 0.468 | 0.086 | 0.217 |  |
| RBSVM (πT =0.9) | 0.428 | 0.105 | 0.162 | 0.468 | 0.116 | 0.176 |  |
|  |  |  |  |  |  |  |  |

SCORE CALIBRATION