Politecnico di Torino

Language Detection

Machine Learning and Pattern Recognition

Alessandro Masci, s308579

January 2023

Introduction

The aim of this project is the development of a classifier which is able to discriminate utterances spoken in a target language or not. The target language considered is Italian and it is associated with the class label 1, while the non-target utterances are associated with the class label 0. It's clear that this is a binary classification problem.

The input embeddings are 6-dimensional, continuous-valued vectors belonging to both classes and they don't have a physical interpretation.

The training set is made up of 2371 samples (400 belonging to the target class and 1971 belonging to the non-target class) while the evaluation set contains 4403 samples (800 from the target class and 3603 from the non-target class).

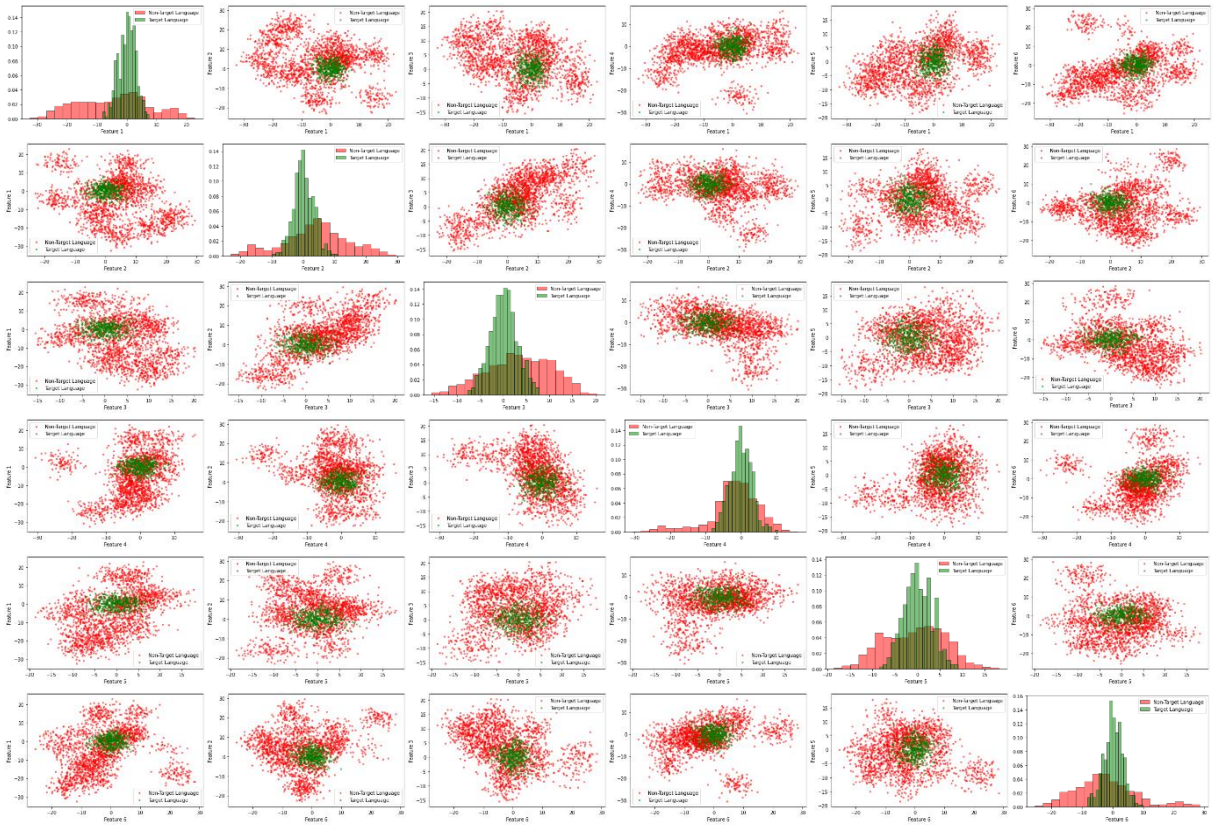
The application has two possible working points: (π=0.1, Cfn=1, Cfp=1) and (π=0.5, Cfn=1, Cfp=1).

In the next sections of this report, different aspects of the problem will be studied, different models will be evaluated and all the results will be displayed together with the different values assigned to each model's parameters.

Dataset visualization

This section is useful to visualize the distribution of the different attributes for the different classes. In particular, as can be seen in [1], some histograms and scatter plots have been plotted.

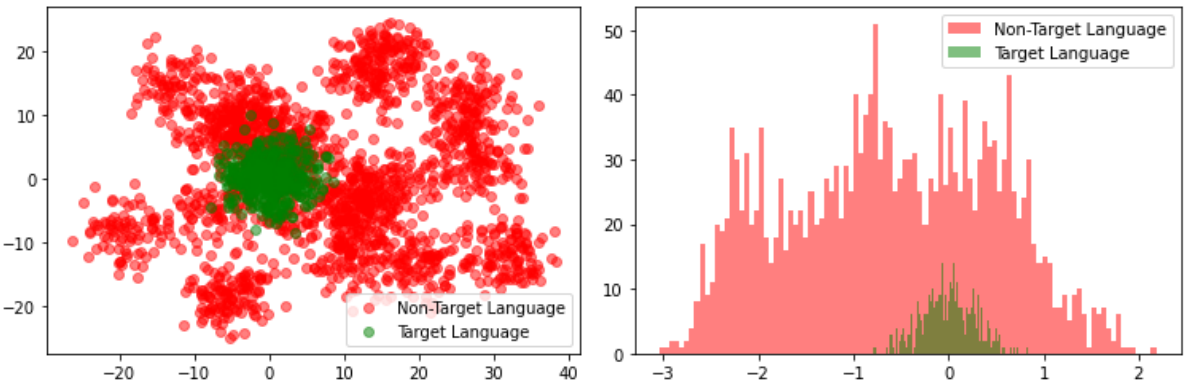
The scatter plots are useful to visualize pairs of values for each class, in the first row of [1], for example, the scatter plots have feature 1 on the x-axis and each of the remaining features (except feature 1) on the y-axis.



[1]

On the main diagonal of [1], have been plotted 6 different histograms, each of them showing a specific feature of the dataset and can be noticed that the target class features can be approximated with a Gaussian distribution while it is not possible for the non-target class features.

To further analyze the features of the dataset, 2-D PCA and LDA have been applied to the training set. In [2] can be seen the plots obtained.



[2]

The PCA plot shows that the non-target class features and the target class features overlap and, while the target class features are arranged in a singular cluster, the non-target class features are distributed in more clusters.

The LDA plot, instead, highlights two different aspects of the dataset: the first one is the huge difference between the number of samples of the target class and the non-target class; the second aspect is the impossibility of linearly separating the features of the two different classes and this aspect will be more evident during the evaluation of the different models in the next section.

Validation phase

In this part of the work different approaches, models and techniques will be evaluated on the training set in order to choose the best model and the best model's parameters to be used during the evaluation phase.

In the beginning, generative models were exploited, in particular:

- Multivariate Gaussian Classifier;
- Naive Bayes Gaussian Classifier;
- Multivariate Gaussian Classifier with Tied Covariance;

Then two other types of generative models will be discussed:

- Linear Logistic Regression;
- Quadratic Logistic Regression;

After the generative models, the focus will be moved towards SVM and GMM and the correspondent results will be displayed.

To choose the best model and evaluate them fairly, the comparison will be based on the value of the primary metric. This metric is defined as the average actual costs of the two working points.

The validation phase is done using a k-fold approach (with k=5) in order to find a training set and an evaluation set starting from the initial training set. In this way is possible to evaluate the performances of all the models and choose the best values of the hyper-parameters.

Gaussian classifiers

These models were evaluated using different values of PCA up to 6 and were evaluated in both the working points in order to estimate the primary metric.

In the above table [3] the results can be seen:

Classifier	PCA	minDCF ($\pi = 0.1$)	minDCF ($\pi = 0.5$)	Primary Metric
MVG	None	0.509	0.135	0.322
	2	0.735	0.195	0.465
	3	0.624	0.140	0.382
	4	0.566	0.141	0.353
	5	0.516	0.134	0.325
	6	0.509	0.135	0.322
Naive Bayes MVG	None	0.546	0.143	0.344
	2	0.736	0.193	0.464
	3	0.626	0.143	0.384
	4	0.571	0.140	0.355
	5	0.512	0.131	0.322
	6	0.514	0.132	0.323
MVG Tied Covariance	None	1.0	0.629	0.814
	2	1.0	0.584	0.792
	3	1.0	0.606	0.803
	4	1.0	0.610	0.805
	5	1.0	0.632	0.816
	6	1.0	0.629	0.814

[3]

The best results are highlighted in different colours and the best configurations are the one without PCA using the MVG and the one using Naive Bayes MVG with PCA equal to five.

The first thing to be noted is that the results with PCA equal to 6 or without PCA are the same (except for Naive Bayes MVG), so, from now on, PCA will be evaluated only for values up to 5.

From these results, it can be seen that the MVG and the Naive Bayes MVG produce similar results while the MVG with Tied Covariance perform worse.

The results obtained with MVG and Naive Bayes MVG are not outstanding and probably this is due to the distribution and the correlation of the different features of the training set samples. As can be seen in [1], the target class samples are distributed like a Gaussian curve while the no-target class samples don't follow a Gaussian distribution.

Moreover in [1] and [2] can be noticed that the features are strongly dependent on each other and this could be a problem for the Naive Bayes assumption of independent features within a class.

Concerning the MVG with Tied Covariance is worse and this is because the assumption that the classes share the same covariance matrix does not suit well the training set and it is reflected in bad results in terms of the primary metric.

Binary Logistic Regression

In this section, the binary version of the logistic regression is implemented to discriminate between target language and non-target language embeddings. Logistic regression is implemented using

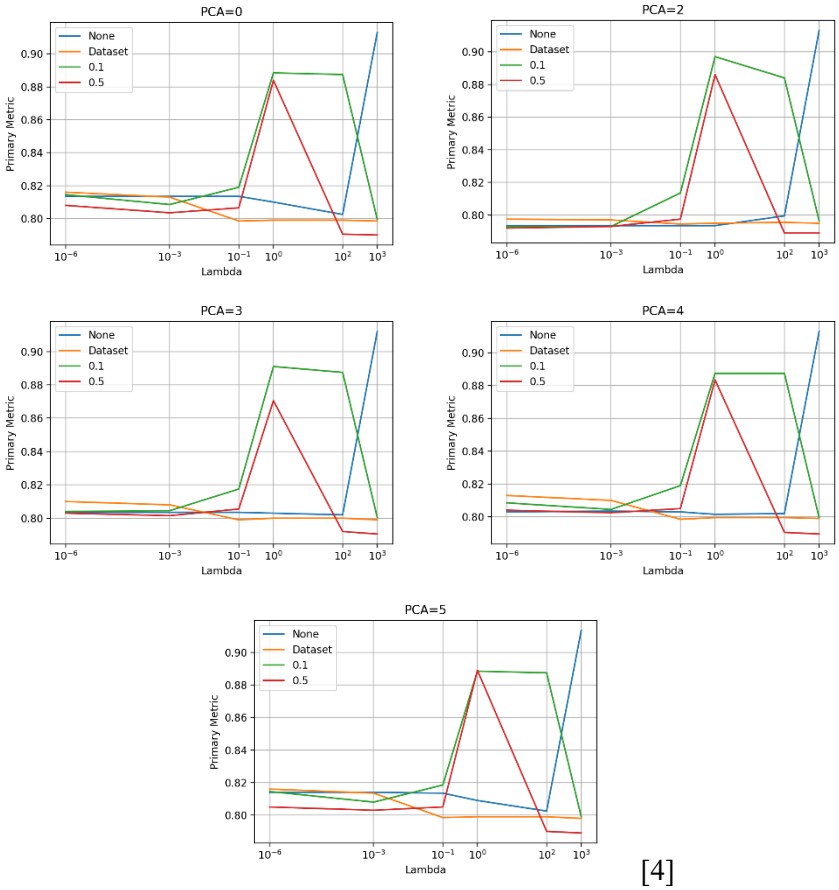
$$J(\boldsymbol{w}, b) = \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-z_i(\boldsymbol{w}^T \boldsymbol{x}_i + b)}\right), \quad z_i = \begin{cases} 1 & \text{if } c_i = 1 \\ -1 & \text{if } c_i = 0 \end{cases} \quad \text{(i.e. } z_i = 2c_i - 1\text{)}$$

Where λ is a hyper-parameter of the model useful to regularize the results.

During this phase, different aspects were evaluated. First of all, a PCA with values from 0 (i.e. without PCA) to 5 was applied and for each value of the PCA, different values of λ were used in order to compute the primary metric in both the working points.

The λ values used are $1 * 10^{-6}$, $1 * 10^{-3}$, $1 * 10^{-1}$, 1.0, $1 * 10^{2}$ and $1 * 10^{3}$; the prior probabilities of the target language class evaluated are the ones from the working points (0.1 and 0.5) but also the prior probability of the dataset were evaluated. Lastly, a linear regression without prior probability was used.

In [4] can be seen the results obtained with the different configurations in terms of the primary metric:



[4]

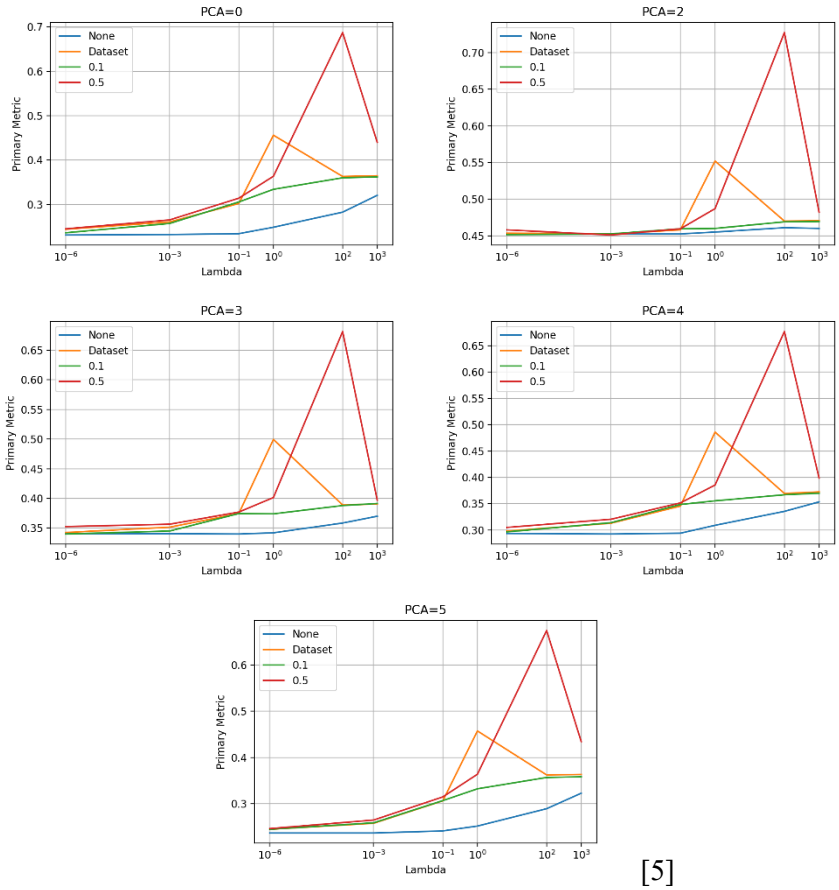
It can be seen that the value of PCA does not affect so much the value of the primary metric while the prior probability and the λ value strongly change the outcome.

The best value of the primary metric obtained is 0.789. This is not a good result and this is due to the fact that the samples in the training set are not linearly separable so it's very hard to find a good linear separation rule.

Since a linear regression does not perform well, a quadratic logistic regression has been evaluated using the above feature expansion process:

$$\phi(x) = \begin{bmatrix} vec(xx^T) \\ x \end{bmatrix}$$

The same parameters used in linear regression were used also for quadratic linear regression. In [5] the results are shown:



[5]

The results obtained are better than the ones obtained with linear regression, as expected. The configurations with the best results are reported in the table below:

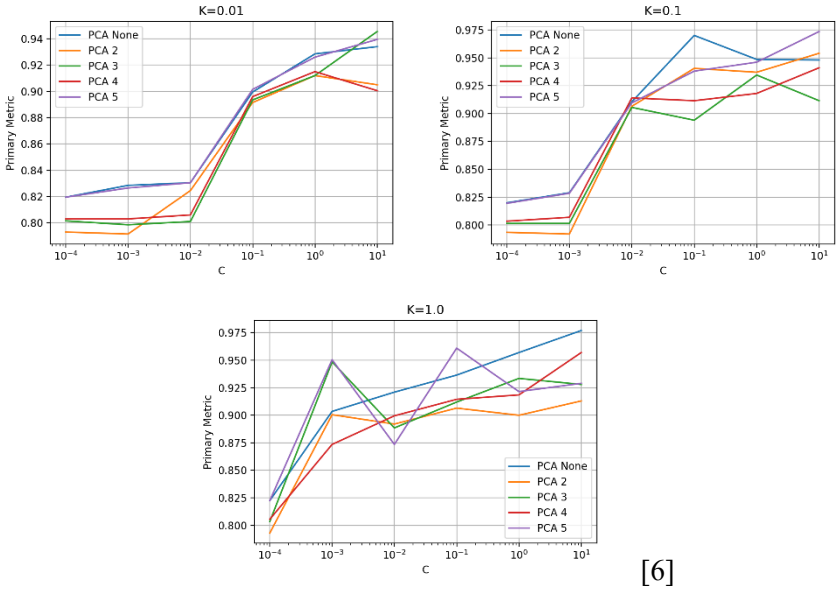
Lambda	Prior probability	PCA	minDCF ($\pi = 0.1$)	minDCF ($\pi = 0.5$)	Primary Metric
1e$^-$6	None	None	0.356	0.106	0.2309

In conclusion, quadratic logistic regression performs really better than linear logistic regression and this is due to the non-linear distribution of the samples in the training set. Moreover, comparing the quadratic linear regression results with the Gaussian classifiers results, the first one is better in terms of primary metric.

Support Vector Machines

This section shows the results obtained using the SVM.

The first part shows the outcome using a Linear SVM with different values of the hyper-parameters. The graphs in [6] show the results with K equal to 0.01, 0.1 and 1.0. In each graph can be seen the results with different values of PCA from 0 (i.e. without PCA) to 5.



[6]

For a value of K equal to 1.0 the values of the primary metric are not good while they're better using K equal to 0.1 and, in particular, equal to 0.01.

Can be also noted that high values of C result in worse performances whatever the value of K and PCA. The best values of C are between $1 * 10^{-4}$ and $1 * 10^{-3}$.

Concerning the values of PCA, the best results is always achieved with a PCA equal to two, more in particular the best outcome has been obtained with the following configuration:

K	C	PCA	Primary Metric
0.01	0.001	2	0.7915

This result is not better than the one obtained with the linear regression that, since now, is the best one achieved during this study.

In order to improve the performances of the SVM, different kernels have been analysed.