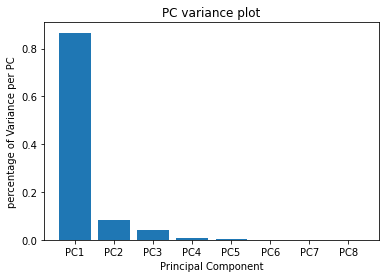
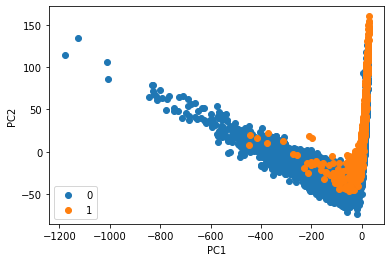
**PCA**

First of all we tried to see if the class separation could be helped by appling a PCA step on the raw data.



We isolated only the first two principal components because the algorithm identified them as the ones with the greatest variance.



But as we can see this method didn’t enanched the class separation, so we decided not to use this pre processing method.

**GAUSSIAN CLASSIFIERS**

Testing Full Covariance, NaiveBayes and Tied Covariance Classifier it highlights Naive Bayes and Full Covariance classifiers perform worst w.r.t. Tied Covariance Classifier on raw data. We used for our first evaluations a single fold and then compared the results with a 5-fold approach dataset with different pre-processing method.

|  |  |  |  |
| --- | --- | --- | --- |
| SINGLE FOLD | **Raw Data** | **Gaussianized Data** | **Z-Normalized Data** |
| ***Full Covariance*** | 0.041 | 0.083 | 0.043 |
| ***Naive Bayes*** | 0.063 | 0.058 | 0.063 |
| ***Tied Covariance*** | 0.024 | 0.060 | 0.024 |

|  |  |  |  |
| --- | --- | --- | --- |
| 5-FOLD | **Raw Data** | **Gaussianized Data** | **Z-Normalized Data** |
| ***Full Covariance*** | 0.039 | 0.078 | 0.039 |
| ***Naive Bayes*** | 0.062 | 0.057 | 0.062 |
| ***Tied Covariance*** | 0.022 | 0.055 | 0.022 |

Analyzing the obtained results we infer that a 5-fold approach give us better overall result on errors. Now on we are going to evaluate our models only with a 5-fold approach.

We chose to leave behind the Naive Bayes because gave us the worst result error wise. Then we chose to evaluate our classifier on different application, also taking account of sensitivity and specificity because after consulting confusion matrix we found our dataset was unbalanced.

|  |  |  |
| --- | --- | --- |
| CM | 0 | 1 |
| 0 | 8058 | 152 |
| 1 | 46 | 669 |

Example CM of Tied Covariance 5-fold on app (0.5,1,1)

Then we evaluated DCF on different applications:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Raw Data | **(0.5, 1, 1)** | | **(0.1, 1, 1)** | | **(0.9, 1, 1)** | |
| **Error | DCF**  **Sensitivity | Specificity** | | | | | |
| ***Full Covariance*** | 0.039 | **0.161** | 0.032 | 0.338 | 0.050 | 0.954 |
| 0.868 | 0.970 | 0.848 | 0.979 | 0.899 | 0.955 |
| ***Tied Covariance*** | **0.022** | **0.191** | 0.025 | 0.270 | 0.022 | **1.433** |
| 0.815 | 0.994 | 0.769 | 0.996 | 0.842 | 0.992 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Gaussianized Data | **(0.5, 1, 1)** | | **(0.1, 1, 1)** | | **(0.9, 1, 1)** | |
| **Error | DCF**  **Sensitivity | Specificity** | | | | | |
| ***Full Covariance*** | 0.078 | 0.177 | 0.039 | 0.399 | 0.148 | **0.868** |
| 0.899 | 0.924 | 0.876 | 0.970 | 0.921 | 0.845 |
| ***Tied Covariance*** | 0.055 | **0.140** | 0.024 | 0.246 | 0.168 | 0.607 |
| 0.912 | 0.948 | 0.865 | 0.988 | 0.952 | 0.820 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Z normalized | **(0.5, 1, 1)** | | **(0.1, 1, 1)** | | **(0.9, 1, 1)** | |
| **Error | DCF**  **Sensitivity | Specificity** | | | | | |
| ***Full Covariance*** | 0.039 | **0.161** | 0.033 | 0.338 | 0.050 | 0.954 |
| 0.868 | 0.970 | 0.848 | 0.979 | 0.899 | 0.955 |
| ***Tied Covariance*** | ***0.022*** | ***0.191*** | 0.025 | 0.270 | 0.022 | **1.433** |
| *0.815* | *0.994* | 0.769 | 0.996 | 0.842 | 0.992 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PCA | **(0.5, 1, 1)** | | **(0.1, 1, 1)** | | **(0.9, 1, 1)** | |
| **Error | DCF**  **Sensitivity | Specificity** | | | | | |
| ***Full Covariance*** | 0.074 | **0.185** | 0.047 | 0.482 | 0.190 | 0.783 |
| 0.886 | 0.930 | 0.810 | 0.968 | 0.935 | 0.798 |
| ***Tied Covariance*** | *0.036* | *0.198* | 0.030 | 0.322 | 0.068 | **1.126** |
| *0.825* | *0.978* | 0.754 | 0.992 | 0.882 | 0.937 |

As supposed in the relative paragraph, PCA didn’t improve our estimate. The Tied Covariance model with gaussianized data obtains the best performance in 5-fold cross validation protocol providing the lowest actDCF but a slightly higher error rate compared to the same classifier on **raw** and **z-normalized** data.

Overall the best candidate is the MVG model with Tied Covariance matrices.

**LOGISTIC REGRESSION**

After we gave to the algorithm raw data, we obtained overflow errors which didn’t allow us to proceed with analysis. So we transformed our data using Z-normalization and gaussianization.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Gaussianized Data | **(0.5, 1, 1)** | | **(0.1, 1, 1)** | | **(0.9, 1, 1)** | |
| **Error | DCF**  **Sensitivity | Specificity** | | | | | |
| ***Lambda = 0*** | 0.022 | 0.160 | 0.050 | 0.547 | 0.058 | 0.757 |
| 0.849 | 0.991 | 0.460 | 0.999 | 0.922 | 0.944 |
| ***Lambda = 0.1*** | 0.041 | 0.425 |  |  |  |  |
| 0.577 | 0.997 |  |  |  |  |
| ***Lambda = 0.01*** | 0.026 | 0.215 |  |  |  |  |
| 0.791 | 0.992 |  |  |  |  |
| ***Lambda = 0.0001*** | 0.0222 | 0.167 |  |  |  |  |
| 0.842 | 0.992 |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Z-normalized Data | **(0.5, 1, 1)** | | **(0.1, 1, 1)** | | **(0.9, 1, 1)** | |
| **Error | DCF**  **Sensitivity | Specificity** | | | | | |
| ***Lambda = 0*** | 0.021 | 0.183 | 0.029 | 0.315 | 0.028 | 0.865 |
| 0.822 | 0.994 | 0.704 | 0.997 | 0.906 | 0.979 |
| ***Lambda = 0.1*** | 0.04 | 0.422 |  |  |  |  |
| 0.578 | 0.999 |  |  |  |  |
| ***Lambda = 0.01*** | 0.028 | 0.268 |  |  |  |  |
| 0.736 | 0.996 |  |  |  |  |
| ***Lambda = 0.0001*** | 0.022 | 0.193 |  |  |  |  |
| 0.812 | 0.994 |  |  |  |  |

PCA OVERFLOW!