# Machine Learning I
# Practice Session I

## 1 Goals

The goal of this assignment is: (a) to learn how to use the Principal Component Analysis (PCA) technique presented during the class and (b) to get you acquainted with cases in which PCA can perform well and cases when it does not.

The document is divided into three parts:

- Part 1 – Getting started: Instructions on how to download and run MlDemos.
- Part 2 – How-to: Instructions on how to perform the objectives of the first practicals: importing a dataset, visualizing data, performing PCA
- Part 3 – Tasks and Questions: Set of tasks to be performed during the practical and questions you must answer.

For this first practical you will focus on loading various real-life datasets, visualizing the data distribution and then selecting good projections of the data through Principal Component Analysis. A good projection is such that it improves the separability of the data or reduces the dimensionality of the dataset, or do both. Throughout the practical session, you will be working on the following real data:

1. Wine cultivar
2. Autonomous driving
3. Fault detection of metal plates
4. Faces

You can download and find a description of the datasets on the moodle website for the class here.

### 1.1 Getting started

#### 1.1.1 Using Virtual Machines

1. Login at the terminal with your GASPAR account
   If you are not already in a virtual machine:
   (a) Open the VMware horizon client and login with the GASPAR account
   (b) Select the STI Windows 10 Virtual Machine
2. Download and extract MLdemos under the Documents folder.
3. Launch mldemos.exe

### 1.1.2 Installation on other Windows machines

If you are not using the STI-Windows 10 virtual machine,you can download ML demos to your Windows machine, **download MLDemos**. All you need to do is to:

1. Get the software (downloadable on Moodle or at `http://lasa.epfl.ch/teaching/lectures/ML_Msc/MLDemos-Windows-Latest.zip`)
2. Unzip it in the folder of your choice[1].
3. Run the executable *mldemos.exe*

The software will provide a graphical interface for visualizing the data and algorithms you will use throughout this year.

**Important:** The only maintained and updated version of MLDemos is the Windows version. If your personal computer is a mac or linux, you have to use the virtual machine which can accessed remotely with any OS at `http://vdi.epfl.ch`).

# 2 How-to:

**Numerical datasets**

## 2.1 Import numerical dataset

The numerical datasets used for the practicals are csv files where each row represents a sample, each column an attribute and the last column is the class in which the sample belongs to. Also only the first row of the csv contains a short id tag for each attribute. Feel free to open the csv files with applications like Excel in order to familiarize yourself with the layout.

**Follow these steps to import the datasets.**

1. Launch MLDemos and select *File >Import > Data* from the menu and open the csv file you want to load. In this example we will use the wines' dataset.
2. The data loading interface will pop up (see Figure 1). At this dialogue box select the: "*First row as Header*" option and then click on the : "*Send data*" button. Now you should be able to see how the data are distributed at the first two dimensions where the different colors represent the classes (see Figure 2).

## 2.2 Visualize data

Real life datasets usually consist of many dimensions. Thus a 2D visualization may not be enough to determine if the dataset is easily classifiable and which dimensions are the most important. Thus, we can use different visualizations to interpret high-dimensional data. To achieve that change from 2D view to visualizations at the drop-down menu below the grid (see red box in Fig. 2)

We will focus on two types of visualizations:

- Individual plots which illustrate how the classes are distributed on each dimension using box-plots.
- Scatter plots which illustrate how the samples are distributed on 2D.

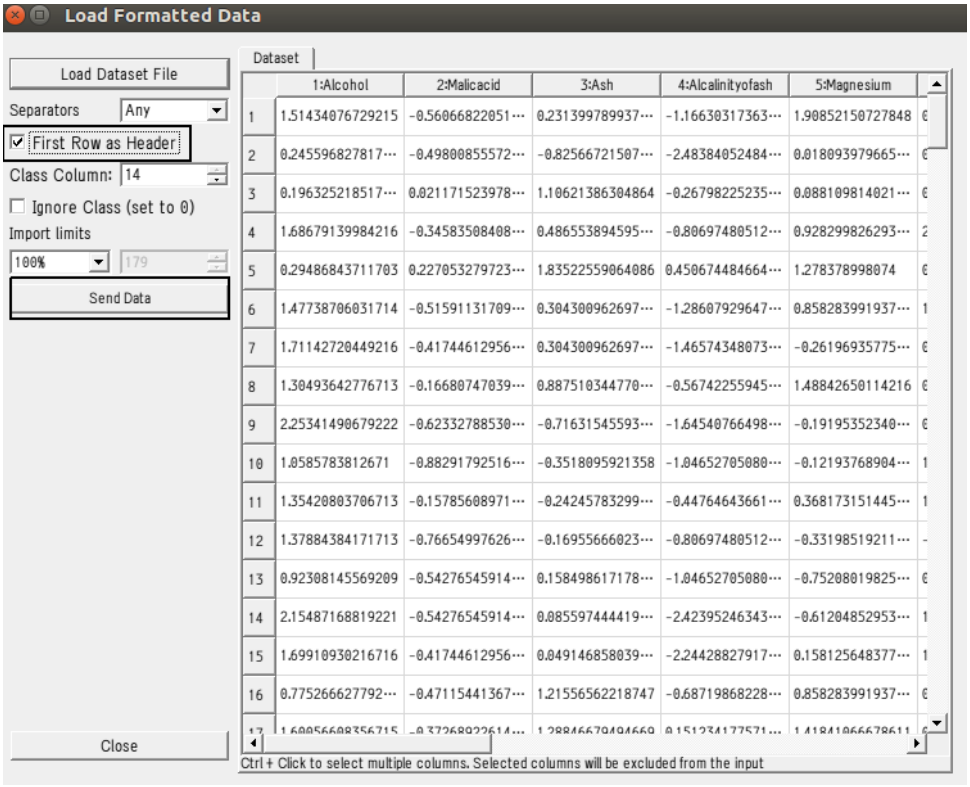



Figure 1: Data loading interface. Select the : "*First row as Header*" option and click on send data button

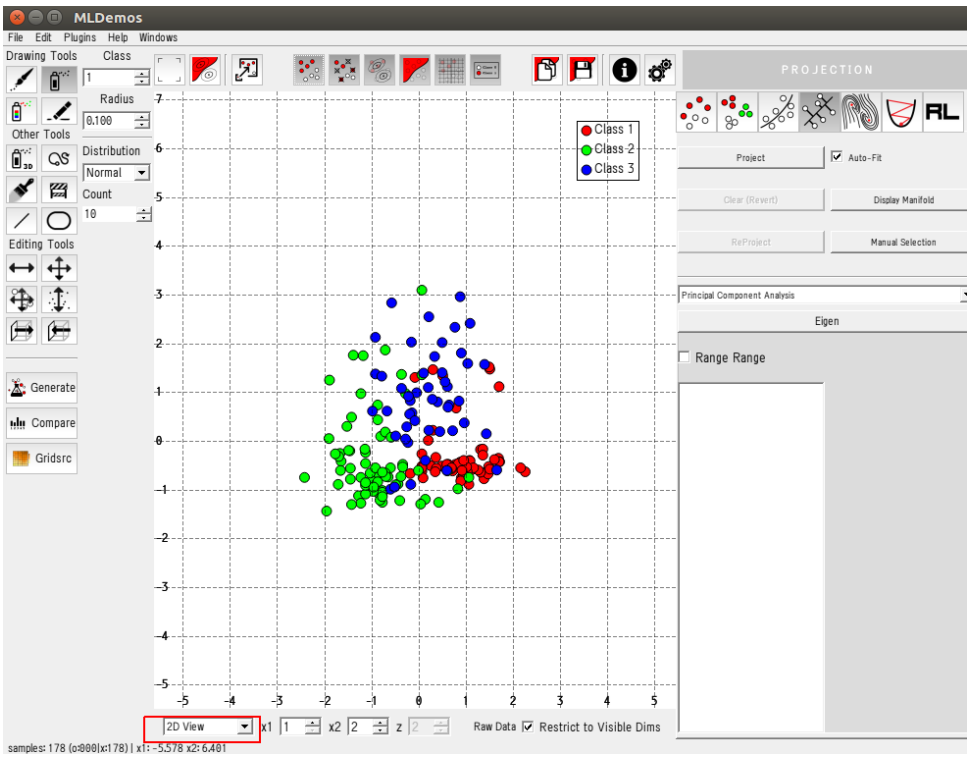



Figure 2: 2D visualization of the dataset.


(a) Individual plots

(b) Scatter plots

Figure 3: Data visualizations

You can select the type of plot from the drop-down menu at the top left (see Fig.3)

## 2.3 Perform PCA and dimensionality reduction

### 2.3.1 Performing PCA:

- Switch to 2D view (Box 1 in Fig. 4).
- From the top right toolbar select projection (Box 2 in Fig. 4 )
- Select the Principal Components analysis from the drop-down menu (Box 3 in Fig. 4 ) and press the project button (Box 4 in Fig. 4 )

This will project the original data to all the principal components. MlDemos also provides the percentage of explained variance for each principal component ( Box 5 in Fig. 4 ) and the reconstruction error curve w.r.t. the principal components ( Box 6 in Fig. 4 )

Reconstruction Error Curve (REC): Projecting onto a subset of principal components results in an information loss which can be measured through the reconstruction error. The more components, the smallest the reconstruction error. The reconstruction error curve (REC) illustrates the rate at which the reconstruction error decreases as one includes more principal components. This creates the curve you can see in box 6 in Fig. 4.

[1]It is advised to decompress the MLdemos zip file in the desktop folder if you are using an EPFL computer to avoid folder/files path issues.

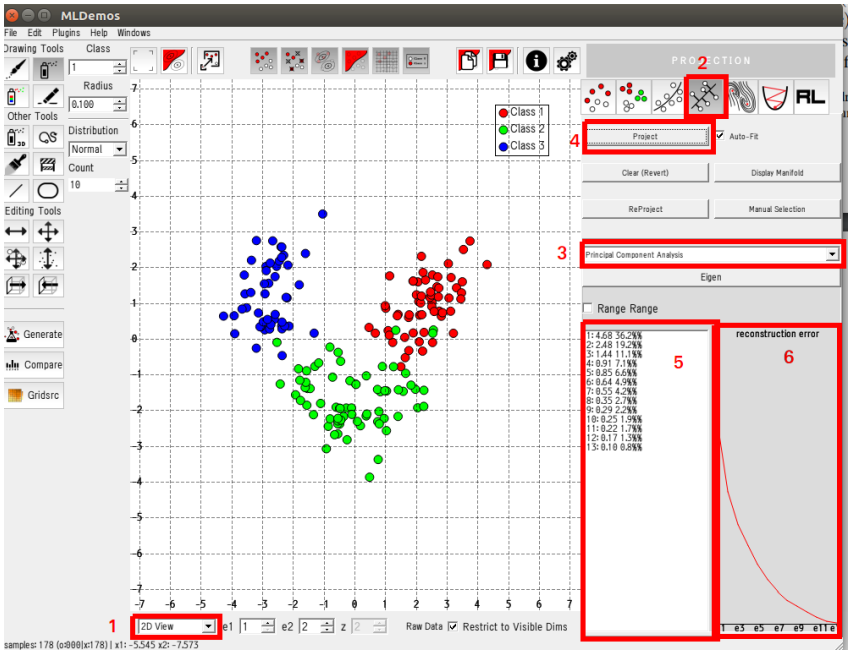



Figure 4: PCA interface.

### 2.3.2 Performing dimensionality reduction:

So far the original datataset is projected to its principal components which are of the same amount as the original dimensions. In order to reduce the dimensions of the data-set, you must select which projections will be kept. In order to do that:

- Tick the range button and select the range of projections which will be kept (Box 1 in Fig. 5)
- Press the ReProject button (Box 2 in Fig. 5).

Now only the projections which were specified by the range are kept and the others are discarded.

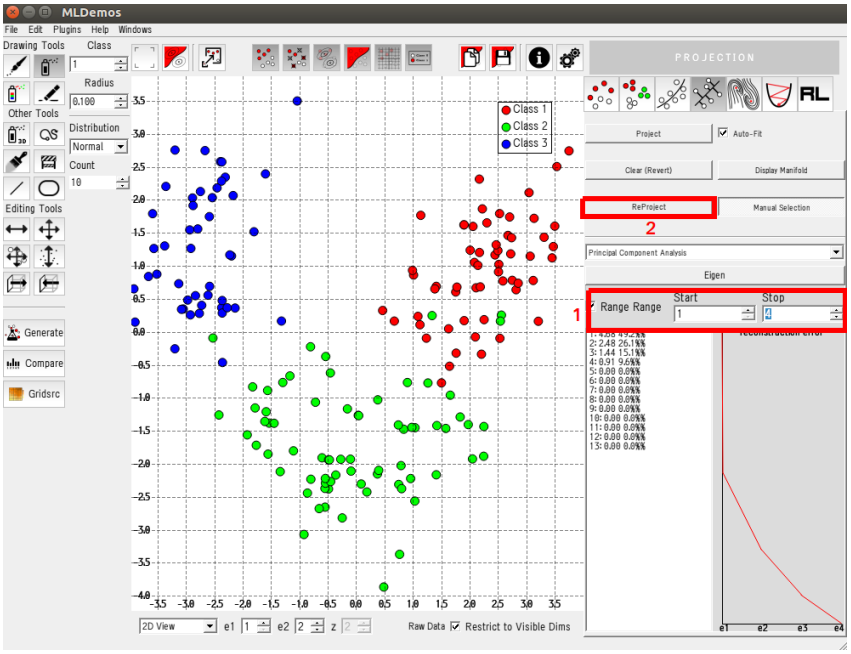



Figure 5: Dimensionality reduction.

**Image Datasets**

## 2.4 Importing and applying PCA on Image datasets

1. Select *Plugins* > *Input/Output* > *PCA faces*
2. From the *PCA faces* click the *Load Dataset* button (Box 1 in Fig. 6) and select the *faces.png* file

   This will load a set of images. The images have class labels, illustrated with a color box around the images. You can visualize the eigenvectors of the data by selecting the Eigenvectors button (Box 2 in Fig. 6).

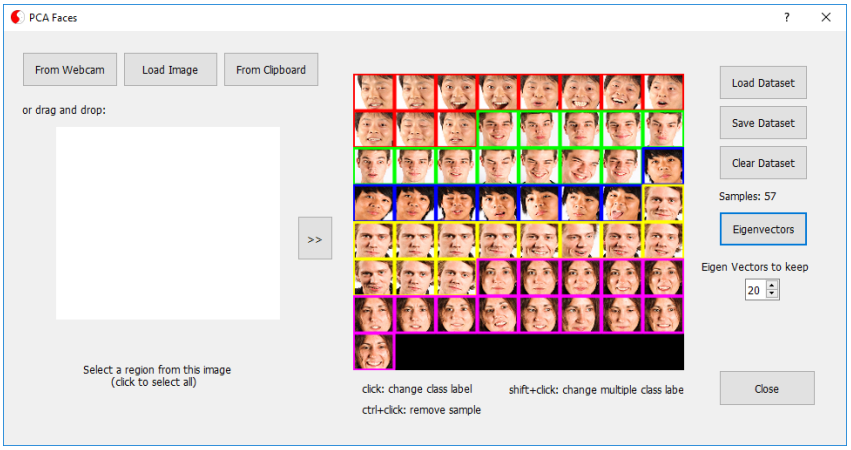



Figure 6: The PCA faces GUI.

Alongside with the visualization of the eigenvectors, you can get the reconstruction error curve and the percentage of explained variance per projection (Fig. 9). Based on those you can

select how many eigenvectors to keep (box 3 in Fig. 6). Just select the number and close the GUI.

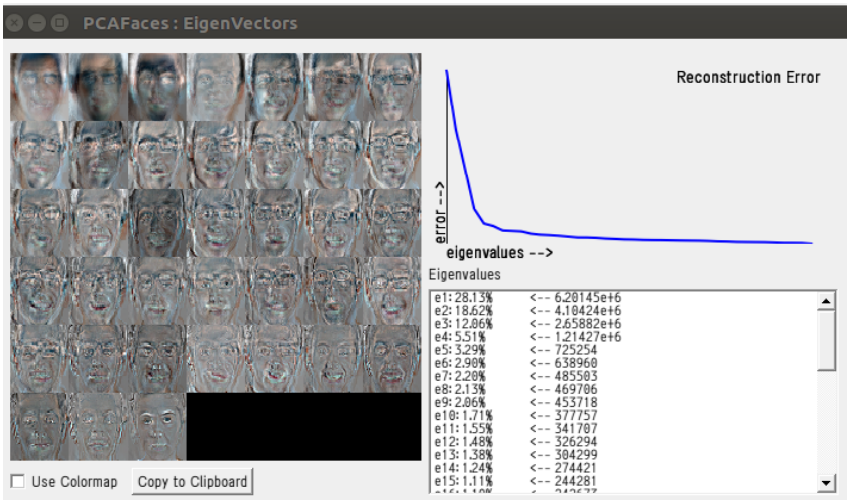



Figure 7: Visualization of eigenvectors as images (eigenfaces).

## 3 Questions

**Make sure to answer in details to each of these questions.**

**Q1:** Import the faces dataset and perform dimensionality reduction. How many and which projections are needed to linearly separate all the classes of the dataset? Which faces (samples) are projected at the highest values on the coordinates of the first three eigenvectors and which are canceled out (projected close to zero on the coordinates of the first three eigenvectors )? Examine the illustration of the eigenvectors (eigenfaces) and explain how they are related.

**Q2**: Import the *Fault detection of steel plates* dataset and visualize the data using individual plots. Which dimensions among the original dimensions are the best to separate the data? Why?

**Q3**: Apply PCA on the *Fault detection of steel plates* dataset and visualize the data using individual plots and 2D scatter plots of the first projections. Do not perform dimensionality reduction. Which projections appear to separate best the classes? Compare the number and quality (in terms of classes' separability) of the best principal projections and best original dimensions. What do you observe?

**Q4**: Import the *Autonomous driving* dataset and perform PCA. Select which components to keep based on the explained variance per component and the reconstruction error curve. Is the decision similar for both criteria? Why?

**Q5**: Select the principal components which allow to separate the classes. Help yourself with the visualization of individual plots. Are those the same as in Q4? Why?

**Q6:** Compare the class separability between the projections on the first three eigenvectors and the projections on the last three eigenvectors. What do you observe and why.

## Solutions

**Q1:** Import the faces dataset and perform dimensionality reduction. How many and which projections are needed to linearly separate all the classes of the dataset? Which faces (samples) are projected at the highest values on the coordinates of the first three eigenvectors and which are canceled out (projected close to zero on the coordinates of the first three eigenvectors )? Examine the illustration of the eigenvectors (eigenfaces) and explain how they are related.

**S1:** All the classes of the dataset are linearly separable along the first two PCA projections (Fig. 8 ). The projection onto the first eigenvector makes the fifth face distinctive (see Dimension 1 in Fig. 10). We can see a similarity between the pictures of the fifth person and the 1st eigenvector (Fig. 9), thus the 1st eigenvector and the samples of the fifth class are correlated which has as result the samples of the fifth class to be projected at the highest values on the coordinates of the first eigenvector. On the other hand the third and forth persons are mapped at the middle values of this eigenvector. This means that they have some features which are captured by the first eigenvector. Finally, the first two faces are projected close to zero at the first eigenvector since it does not represent many of their features. Since the eigenvectors can be interpreted as features of the images, the first eigenvector appears to discriminate the faces based on the hair that appear at the picture. Thus the fifth class is projected at highest values on the coordinates of the first eigenvector. While the second class of images cancels out since it lacks more this feature compared to the other classes.

We can follow the same reasoning for the second and third eigenvectors. The second appears to be a superposition (combination of multiple features) of two or more faces, while the third eigenvector captures mainly the features of the forth face.

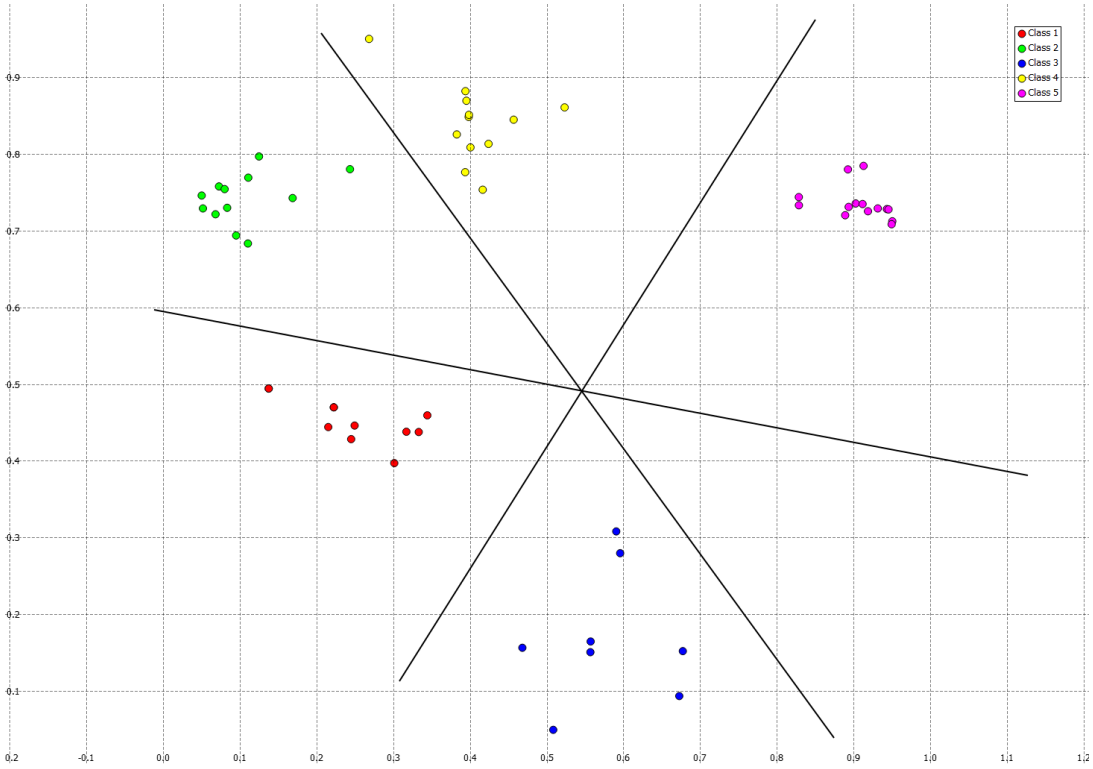



Figure 8: A set of lines which can separate the classes as they are projected at the first two eigenvectors.

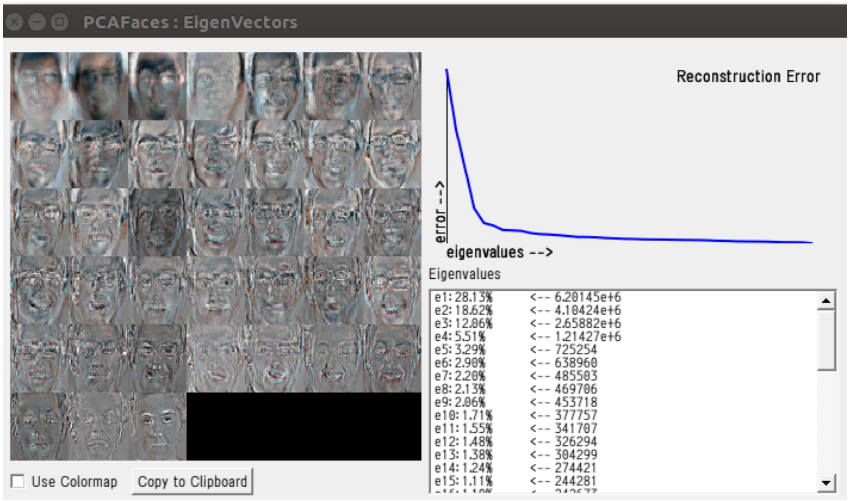



Figure 9: Visualization of eigenvectors as images (eigenfaces).

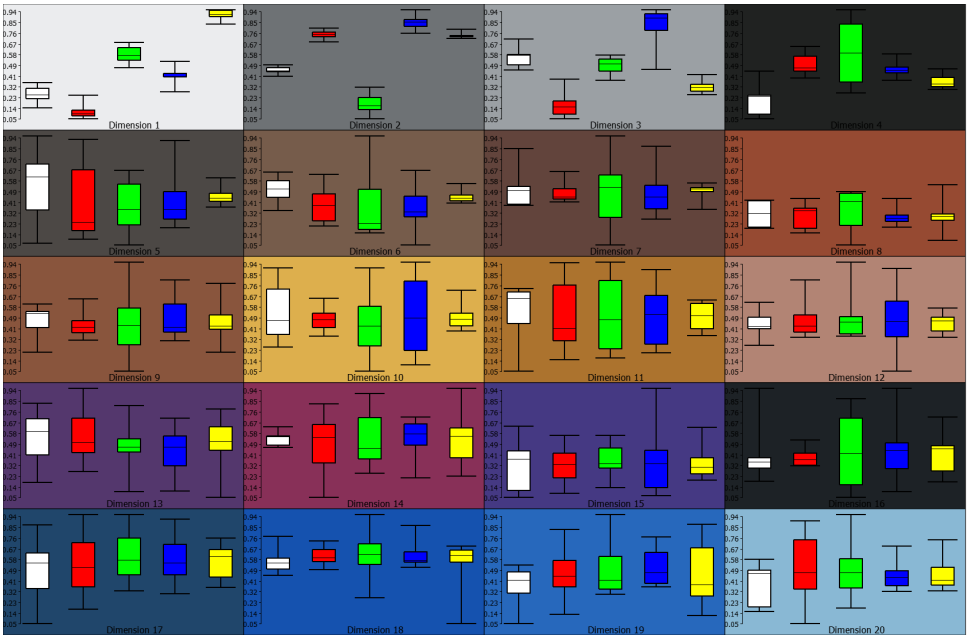



Figure 10: Individual plots of the projected faces dataset.

Figure 11: The faces dataset.

**Q2**: Import the *Fault detection of steel plates* dataset and visualize the data using individual plots. Which original dimensions are the best to separate the classes? Why?

**S2**: The objective of this question is to learn how to interpret the distribution of variance as displayed through the box plots for each class and for each dimension in mldemos, as illustrated in Fig. 12. Each box-plot represents the distribution of one class on a dimension. The range of the box corresponds to the range where 50% of the classes' samples are distributed. The inner line of the box is the mean of the samples. The two outer lines point at the range where 99% of the class is distributed. Therefore, the best

dimensions to separate a class are those where the class distribution is more distinctive compared to other classes. When we look at the distribution of the *original dimensions*, we can observed that a few classes become more separable along some dimensions. For the metal-steel dataset, according to Fig.12, red and green classes are more separable along the first dimension. The distribution of the green class is very distinct along several dimensions, e.g. along dimensions 5, 6, 8, 18s. The blue class's distribution is distinctive in the 22nd dimension. The other two classes seem to significantly overlap at in all dimensions.

We can hence conclude that it is likely that a ML classifier could separate correctly the red, green and blue classes by just using the original dimensions. Classification of the other two classes us likely not feasible. Would applying PCA help? It could help in two ways. 1) It could help separate the two remaining classes by decorrelating their distributions. 2) It could reduce the dimensionality, which would speed up computation of the classifier afterwards. However, if we reduce the dimension, we must make sure that the retained projections do not discard projections that would have de-mixed well the 4 classes.

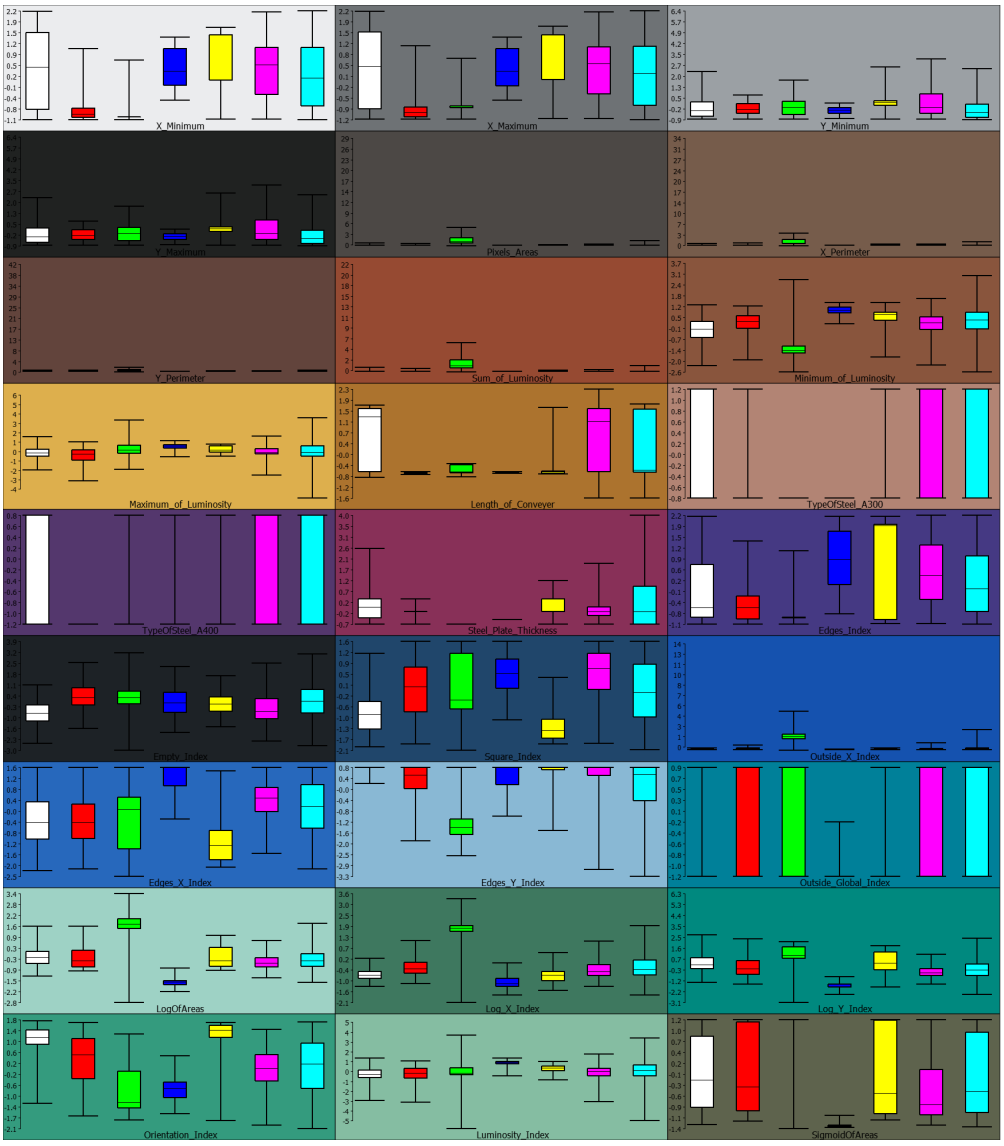



Figure 12: Individual plots of the original fault detection dataset.

Table 1: Structure of a confusion matrix

| Actual Class \ Predicted Class | $C_1$ | $C_2$ | $\cdots$ | $C_c$ |
|---|---|---|---|---|
| $C_1$ | $n_{11}$ | $n_{12}$ | $n_{1\ldots}$ | $n_{1c}$ |
| $C_2$ | $n_{21}$ | $n_{22}$ | $n_{2\ldots}$ | $n_{2c}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $C_c$ | $n_{c1}$ | $n_{c2}$ | $\cdots$ | $n_{cc}$ |

This can be also verified experimentally by using a simple classifier such as k-NN[1], to classify the data, and a confusion matrix[2] to see which classes are mixed. (This was not asked during the practical. We add it to the solutions to help you better interpret which classes are mixed).

K-NN classifier[1]: the k-NN classifier assigns a class to a sample based on a "majority vote" of the classes of its k nearest samples (neighbours). The parameter k is a hyper-parameter which represents the amount of nearest neighbors to be considered. We will see more about kNN during the lectures and practical on classification.

Confusion matrix[2]: The confusion matrix is a $c \times c$ matrix where $c$ is the number of classes. It illustrates which classes are mixed with which by a classification algorithm. You can see the general structure of a confusion matrix in Table1. For example, in the entry $n_{11}$ is the number of samples which actually belong to class 1 and classified in class 1 by the classifier (correct classification). The entry $n_{12}$ is the number of samples which actually belong to class 1 but classified in class 2 from the classifier, which corresponds to missclassification. Thus, if the classes are easily separable a fine-tuned classifier would produce only correct classifications for all the classes. This would result to a diagonal confusion matrix. More details on confusion matrix will be presented on the classification lecture and practical.

A color-coded visualization of the confusion matrix which derives using the k-NN classifier with $k = 15$ is illustrated in Fig.13. The density of the red color represents the amount of samples at each entry. Each entry is represented by a tile in Fig.13. The higher the amount of samples at an entry, the highest the color density of the tile. The order of classes in the confusion matrix is the same as they appear in the individual plots.

Inspecting the confusion matrix we can deduce that the second and third classes (which correspond to the green and blue in Fig. 12 ) are easier separable compared to the others. The second and third rows and columns (bounding boxes in Fig. 13) have their color-density focused mainly on diagonal entries ($n_{33}$ and $n_{44}$) which corresponds to correct classifications. On the other hand, from the confusion matrix can be deduced that the first class is mixed with classes 7 and 8.

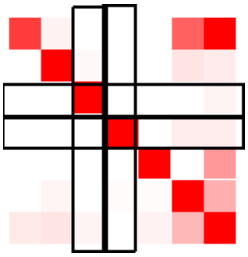


Figure 13: Confusion matrix that derives using the k-NN classifier on the original dimensions of the Fault Detection dataset. The black boxes highlight the two classes with the highest separability.

**Q3**: Apply PCA on the *Fault detection of steel plates* dataset and visualize the data using individual plots and 2D scatter plots of the first projections. Do not perform dimensionality reduction. Which projections appear to separate better the classes? Compare the number and quality (in terms of classes' separability) of the best principal projections and best original dimensions. What do you observe?

**S3**: The projections 1,2 and 3 appear to be the best to separate the classes.The same classes are separated by less principal components compared to original dimensions(examine Figs 14 and 12). For example the green class is distinctive on many of the original dimensions but after PCA it is distinctive only on one projection. This means that there exist original dimensions with redundant information. Also, the same classes that are separable at the original dimensions are also separable at the projections. Thus, the separability of the classes in the dataset has not been increased. This is to be expected since PCA is an unsupervised method i.e. does not have any information regarding the classes of the samples and thus there is no guarantee that the classes will be separable after the projection to the principal components.

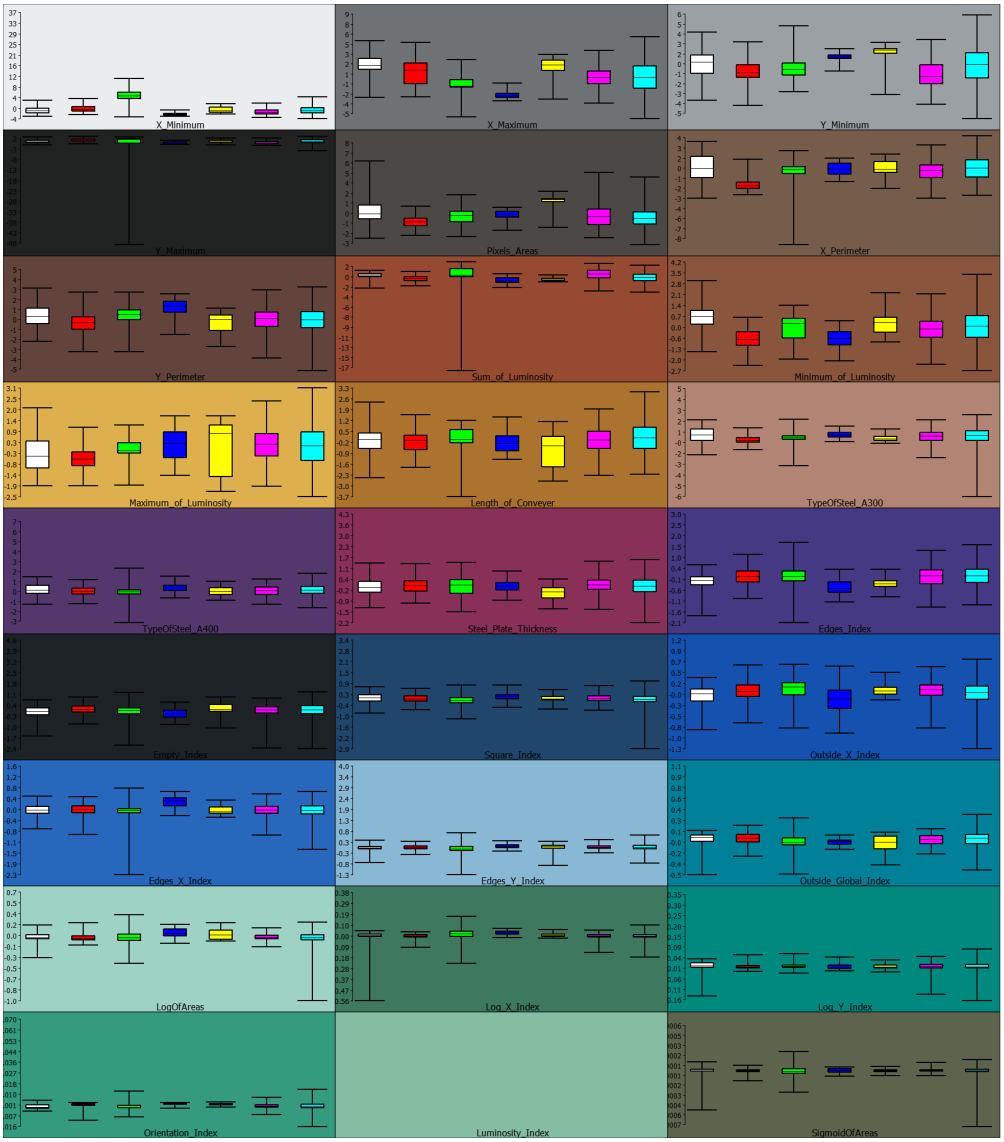
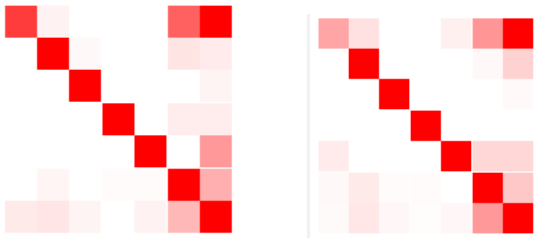



Figure 14: Individual plots of the projected fault detection dataset.

We can also compare the confusion matrices before and after PCA to decide on the impact of PCA on the separability of the classes. The confusion matrices before and after PCA are illustrated in Fig.15. We can easily notice that they are very similar. Thus, in this dataset, the advantage of PCA is to decrease the computational resources that are needed to run the desired classification algorithms by removing redundant dimensions.

**Q4**: Import the *Autonomous driving* dataset and perform PCA select which components to keep based on the explained variance per component and the reconstruction error curve. Is the decision the same for both criteria? Why?

**S4**: For both criteria the elbow method has to be used for deciding on the amount of projections to keep. For example, using the reconstruction error curve, a reasonable amount of components to keep is the point where the curve start to become less steep (see black circle in Fig. 16). Thus, in this example, choosing to keep an amount of approximately 16 eigenvectors appear to be a reasonable choice based on the curve. The decision of how many components to keep has to be similar for both the reconstruction curve and



(a) Before PCA (b) After PCA

Figure 15: Confusion matrix that derives using the k-NN classifier before (left) and after (right) PCA

explained variance since they are equivalent (i.e. maximizing the explained variance is identical as minimizing the reconstruction error ).

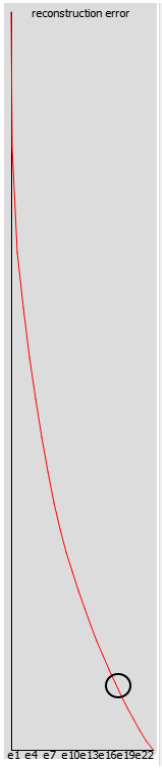
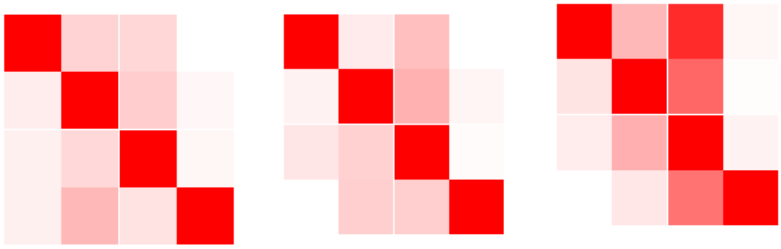



Figure 16: Reconstruction error performing PCA on the autonomous navigation dataset.

**Q5**: Select the principal components which provide the most separable classes using visualization of individual plots. Are those the same as in Q4? Why?

**S5**: If the criterion is the visualization of the classes distributions, then the optimal number of components is different. Using visualization tool, only the first two components appear to provide a separation between classes (Fig. 17). In Q4 we needed 16 eigenvectors based on the reconstruction error curve. This inconsistency happens because the reconstruction error and the percentage of explained variance are unsupervised criteria and so they do not have any information regarding the classes. Thus all the optimal number of components, that derive from the variance and reconstruction error metrics, do not guarantee that



(a) All eigenvectors are kept (b) 16 eigenvectors are kept (c) 2 eigenvectors are kept

Figure 18: Confusion matrix that derives using the k-NN classifier at the autonomous navigation datastet, using the projections on all (left) the first 16 (center) and the first 2 (right) eigenvectors

separate the classes.

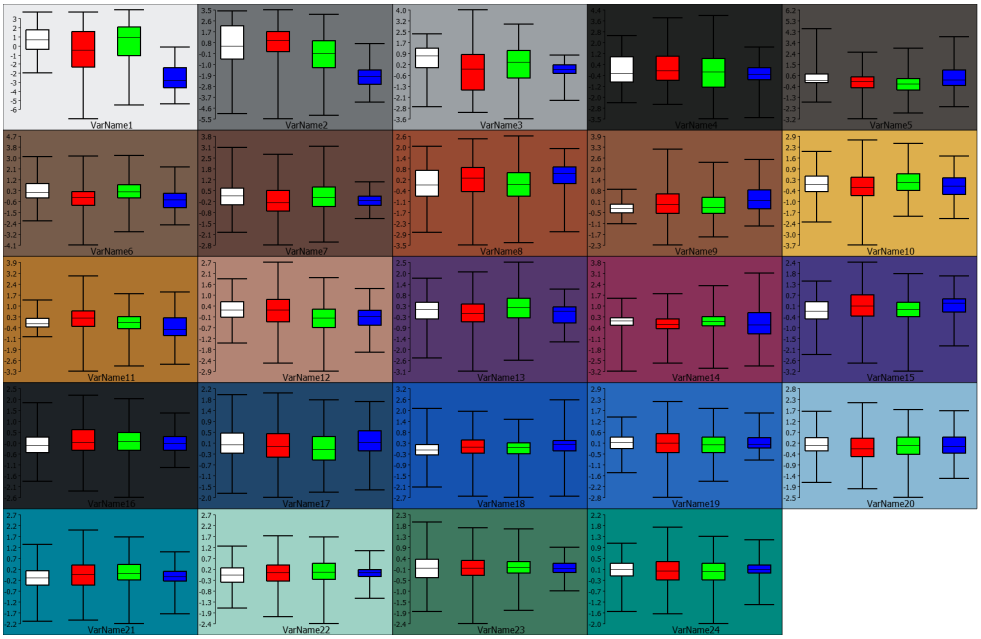



Figure 17: Individual plots of the projected navigation dataset.

Nevertheless, visualizations can be misleading as well since it is hard for humans to anticipate distributions of data in more than 3D. As an example at this dataset, the classes separability gets worse if the projections are drastically reduced (only the first two projections are kept) Fig. 18. Therefore, the best practice is to use the elbow method in order to decide on the number of projections that will be kept.

**Q6:** Compare the class separability between the projections on the first three eigenvectors and the projections on the last three eigenvectors. What do you observe and why.

**S6:** The classes are more separable at the first three components compared to the last three (Fig. 17). This happens because the last eigenvectors have a direction where data vary the least, the contrary holds for the first eigenvectors (given that they are sorted in order of decreased variance). Therefore, is much more likely the classes to be more distinctive at the direction of the largest variance compared to the direction of the lowest.