ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

# Applied Machine Learning Report

Fall 2018

Team members:
Jean Duquenne (SCIPER: 258439)
Thammisetty Devakumar (SCIPER: 285481)

# I.Introduction

This is a report on the practice sessions from the applied machine learning course. The practice sessions gave us the opportunity to apply each of the algorithms seen in class on real datasets generated by us.
The algorithms seen are:

- PCA: The goal is to reduce the dimension of the datasets by choosing the right projection
- Clustering: Assess the performance of K-Means, Soft K-means and DBSCAN
- Classification: Assess performance of kNN, GMM and SVM and select the best method

# II. Datasets

The dataset is made of pens differing by their shape and colors. Each class is made of a pen with different orientation.

1. The first class of objects is composed of eraser which have recognizable feature like their shape and colors, making the clustering easier.
2. The second class is a highlighter which got distinctive color orange.
3. The third is a marker pen which is thin and black, the data from this class is harder to distinguish from 1st two which makes it a good candidate to measure the efficiency of the algorithms to correctly cluster and classify.


Figure 1: The different objects used for the dataset with similar dimensions and orientations

The dimension of the pictures are 3024 x 4032 pixels. The main features to differentiate the classes are the colors and shapes. However, noise is introduced by tilting the object around mean orientation.

## II.a. Building the datasets

The dataset should be relatively easy to cluster with some exceptions to appreciate the quality of a certain solution. We took 17 pictures of the the highlighter, 16 pictures of the marker pen and 16 pictures of the eraser. Different tilts are introduced to simulate noise in the parameters, which is to be filtered with PCA. Further, the mean orientation of the three objects is kept vertical to test the efficacy of clustering algorithm (Easier to separate the feature if all markers are in different orientation compared to highlighter). Also, it is to be noted that orientation is not a feature we are looking for in this exercise, hence it can be treated as noise parameter.

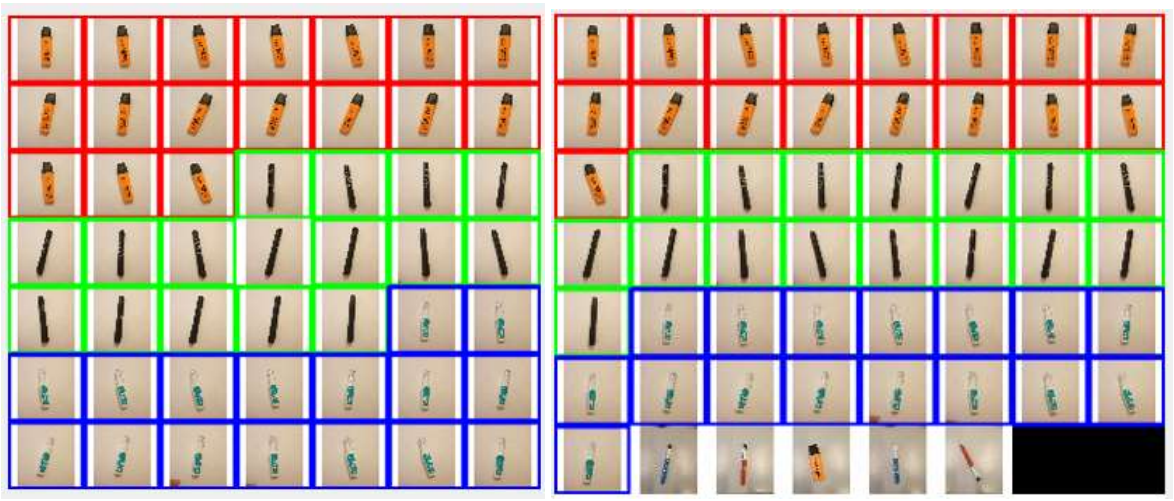

Figure 2: Data sets used for PCA and clustering (Left), Classification (Right)

For classification purpose, more complex data set is preferred to compare various hyperparameters and algorithms. Hence, additional 5 images which have features close to the original dataset are selected(Refer figure 2-right). These are added in order to compare non-linear classifiers as well as to increase the variability among feature space.

# III. Dimensionality Reduction (PCA)

As we see in figure 3, the first three projections contain the main features of all three objects, color and shape. We can observe that the first projection is a highlighter, second projection gives marker parameters and third one is an eraser. Further, it can be seen that the noise parameter (Orientation) start making appearance in the projections from 6th eigenvector.

In terms of eigenvectors, the first three eigenvectors explain more than half of the variance of the data in dataset 1 and separate the classes very well. Though first three eigenvectors represent the main appearance of the three objects and encapsulate approximately two third of the variance of the data, we have chosen first five eigenvectors in dimensionality reduction so that the data loss is minimal. It is to be noted that the introduced noise of the order of 4% is successfully eliminated using PCA in this particular data set with first five eigenvector selection. Moreover, projection 5 is where there is a sharp decrease in the cumulative reconstruction error, hence we select first five projections for further analysis.

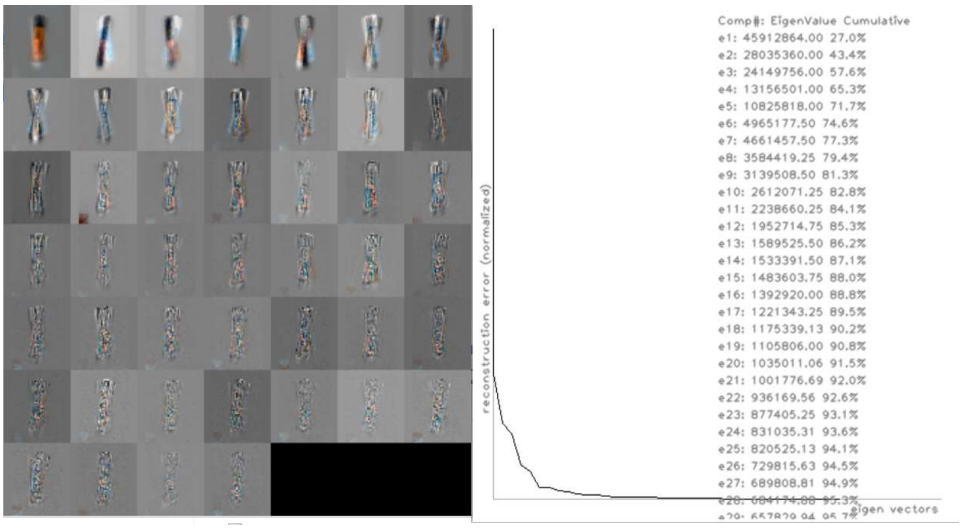


Figure 3: More significant eigenvectors after the PCA and their cumulative reconstruction error.

As we can see below, e1 and e2 make clearly linearly separable clusters as e3 and e5 mix the classes (everything is mixed in exemple: e6e5 or e5e4 or even e3e4) . That is why linear separation is not possible for this last projection. For these reasons, e1 and e2 have been chosen for the clustering. For the first three eigenvectors, the fact that the data-points of the same class are close to each other and far from the other ones shows that it does not seem to be wrong correlation in the projections.

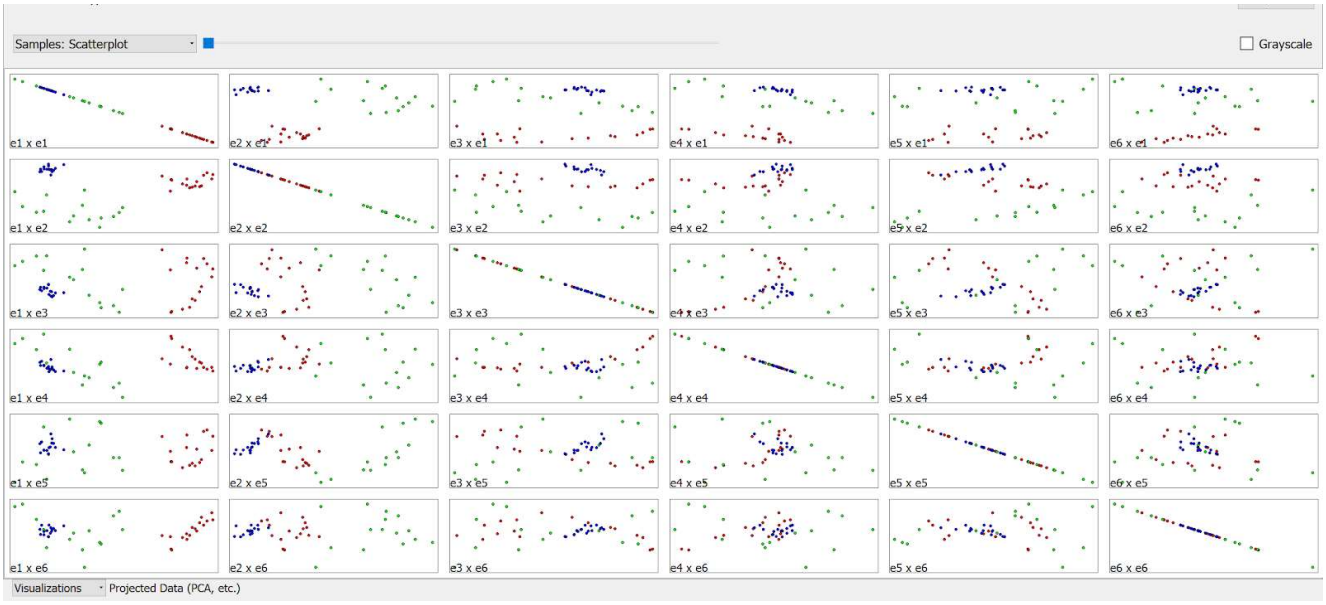


Figure 4 : The ‘scatterplot’ visualisation of the dataset , we can observe that the firsts eigenvector representation separate the classes and eigenvectors above 3: 4,5 and 6 mix the data.

# IV. Clustering

We will try to cluster the data with different algorithms:K-means, soft K-means and DBSCAN. We will assess qualitatively and quantitatively the performance of these methods, reporting the major differences in the metrics used to evaluate the clustering method, each with different values of its hyperparameters.

# IV.a K-means

The initialisation of K-means is random so the results vary from an execution to another. Our green class is spread on sheet (Figure 5) which make the cluster vary from an initialization to another. Indeed the centroids are randomly initialized, several runs of the algorithm can give very different results as we can see in the figure, even if a certain cluster configuration happen most of the time some extreme cases occur. It can also be noticed that the prior knowledge about the number of clusters is essential.

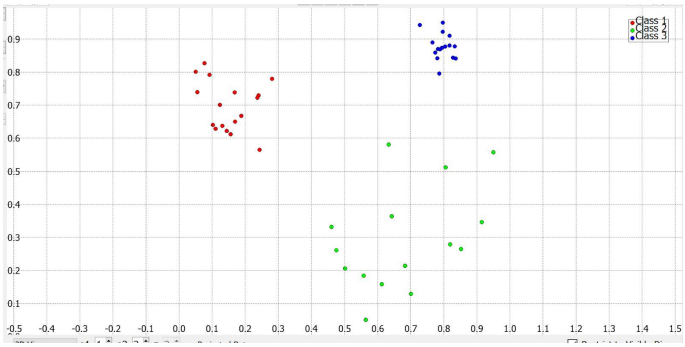


Figure 5: the dataset projection on e1e2.

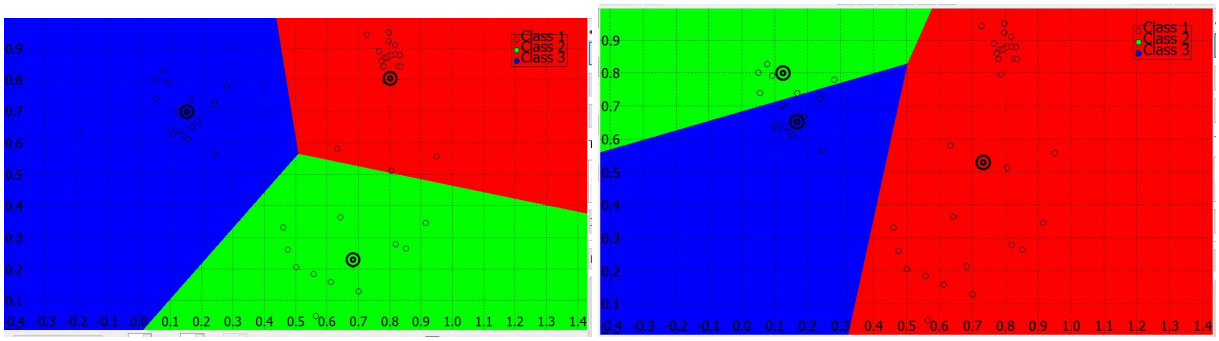


Figure 6: For an euclidean norm (L2) we can get very different cluster, most of the time ( about 80%) we get a cluster similar to the one on the left, in some extreme cases we can get the left cluster.

The distance from the centroid of each datapoint can be computed using different metrics such as L1 norm, Euclidean distance or even infinite norm as shown below. As the function minimises the square sum of the distances, the higher the power of the norm, the most penalizing it is for the algorithm to have several distant points from the centroid. Moreover, a high-power distance leads to a high possible curvature of the separation between the clusters. The L1 norm is an exception because it computes the distance by summing horizontal and vertical displacements, which makes the separation lines being only horizontal, vertical or diagonal at 45°.

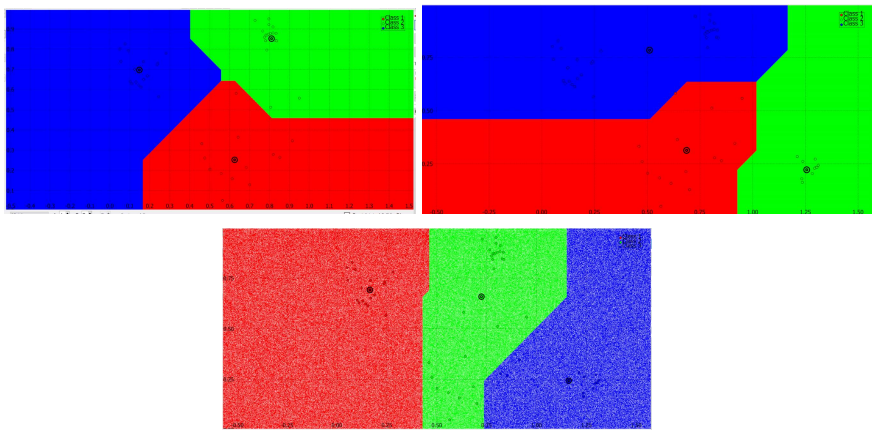

Figure 7: Different cluster execution with L1 which show poor clustering for the two on the right and a decent one on the left .

For L2 an l infinite the quality of the cluster is on average higher and the results are less random , it stills rare for a clustering to accurately cluster the classes.

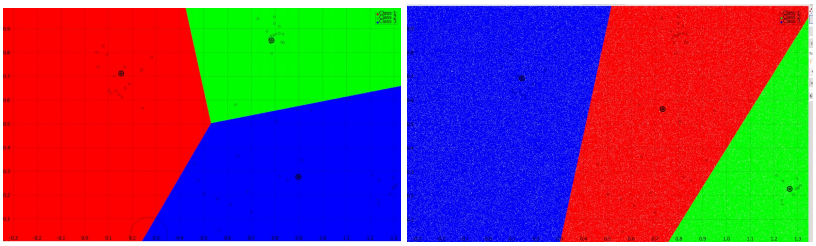

Figure 8: cluster for euclidean distance bad on the right and better on the left.

For L-infinite we finally obtain accurate cluster most of the time as seen below even if we can observe some extreme cases.

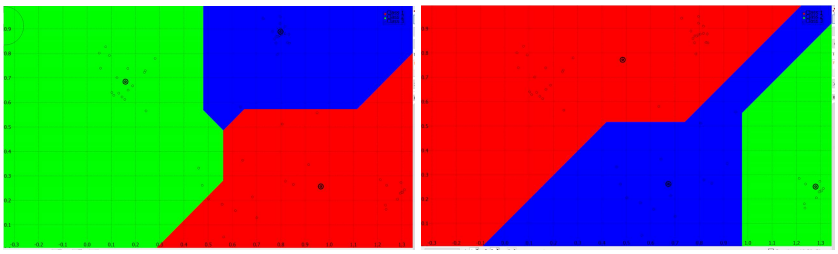

Figure 9: cluster for L infinite with on the left a relatively good cluster and on the right a bad cluster

## IV.b Soft K-means

To have a “softer” partitioning, so a less binary separation between the clusters, the step function used for the clustering decision can be replaced by a sigmoid with a parameter in the exponential named beta. As we can see in the figure below, a too small beta leads to a complete uncertainty concerning the belonging of the data points and a very high beta to a result close to the classical K-means.

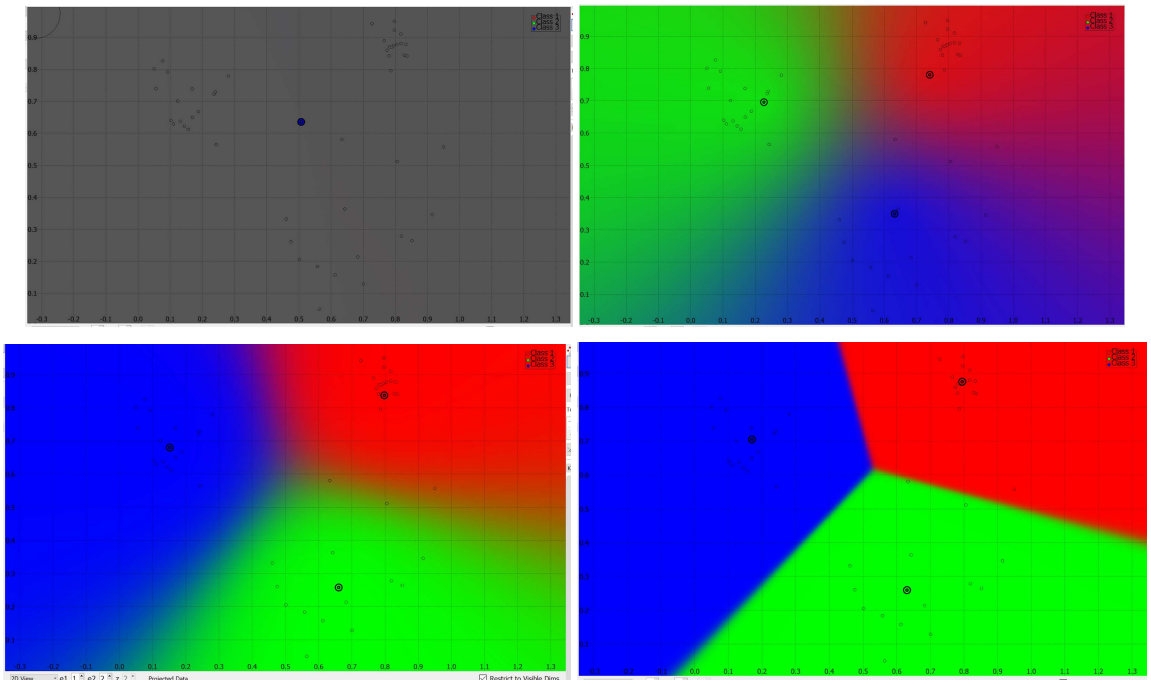

Figure 10: Soft K-means clustering for beta = 1, 5, 10 and the maximum value 100 (respectively)

## IV.c DBSCAN

This algorithm is based on the proximity between data points and cluster them if more than mdata points are in a circle of radius epsilon around each point, otherwise it is considered as outlier.

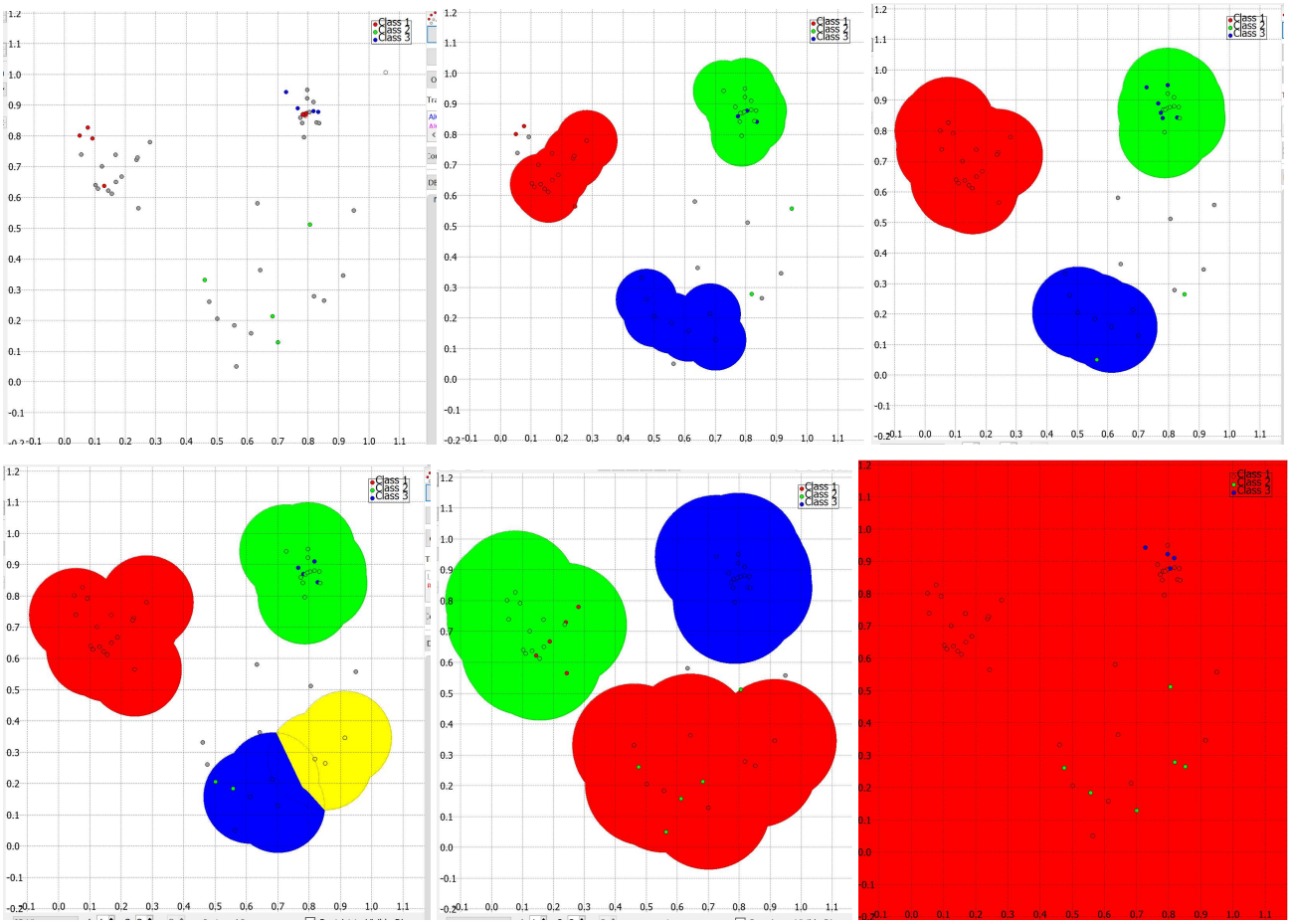

Figure 11: Clustering with DBSCAN with (epsilon, mdata) of: (0.01, 3), (0.1, 3), (0.15, 5), (0.15, 3), (0.2, 3) and (1, 3)

As shown in figure 11, a too small epsilon leads to consider each datapoint as an outlier or a cluster depending if mdata is high or low. With a large epsilon, all the points belong to one single big cluster independently of the value of mdata. If epsilon is a little too small, depending on mdata, some clusters may be split or merged, with a part of their points possibly set as outliers.

## IV.d Quantitative assessment on K-Means and Soft-K-Means.

We first observe the value of BIC and AIC for the BIC and AIC optimization giving most of the time a optimal number of clusters of 3 both for AIC and BIC (the value for both is in the range of -230 -200 for this minimal ), The result are recurrent for all the hyperparameters, the minimum of BIC and AIC is falling in our range with an optimal number of cluster of 3. In some few cases the curves find there minimum in 4 clusters, which is due by the randomness of the initialization of the k-means method. For the F1 optimization we compare the different training labels ratio:

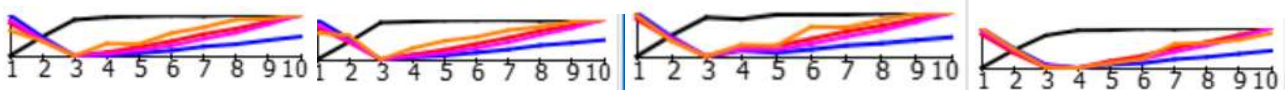

Figure 12: from left to right (method,hyperparameter, training label ratio): soft -means, $\beta = 5$ ,33% ; soft k-means, $\beta = 5$ , 75%; K-means,-,33%; K-means,-,75%. F1 being the black line.

We can observe that the F1 optimization find the optimal number of cluster at 3 with some exception where it find 4.

In conclusion, K-means algorithms make globular clusters with no outliers and are very dependent of a priori knowledge about the number of clusters and the initialization of the centroids of the clusters. However, the convergence is relatively fast and guaranteed. DBSCAN for its part is

more expensive in terms of computation and has more parameters to tune but can generate non-globular clusters and is robust to outliers if there are noisy measurements

# V. Classification

In order to classify the dataset, three different algorithms namely Gaussian Mixture Models with Bayes classifier (GMM), k-Nearest Neighbors(KNN) and Support Vector Machine(SVM) are used. The dimensionality of the data set is reduced using PCA, final set contains only 5 projections with reconstruction error < 5%.

Projection of the data onto e1 and e2 eigenvectors is given below. Since three different classes of objects are selected, the required features are linearly separable and correlated. The optimal selection of hyperparameters, train to test ratios, performance measure and cross validation results are discussed in this section.

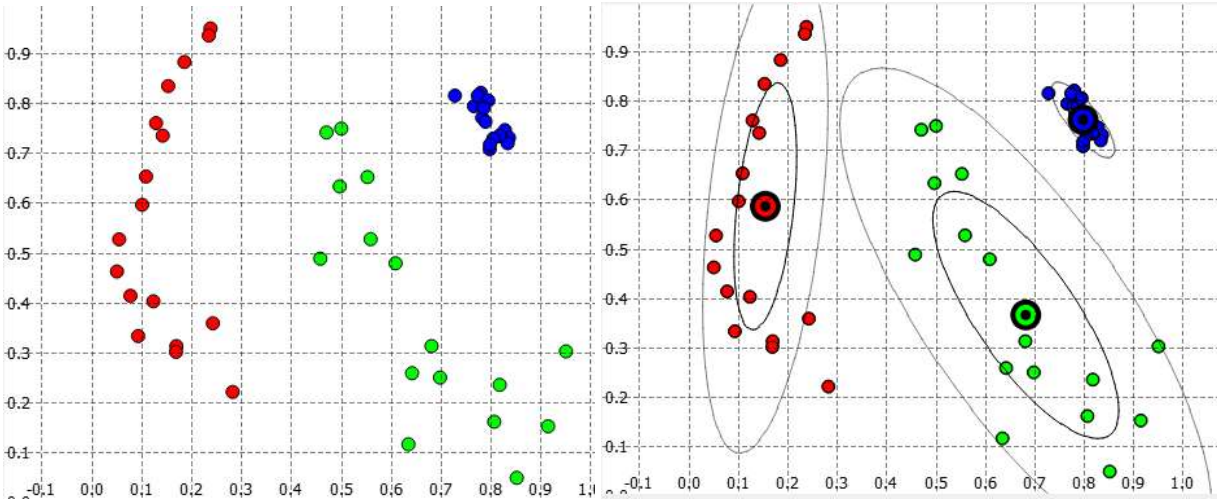

Figure 13: PCA projection of data set (Left), Typical GMM classifier with full covariance (Right).

**Train/test ratio**

Since each class has 16 to 17 samples, test/train ratio of 66% gives 11 data points per class on average for training and remaining (5) data points for testing. Having lower test/train ratios leads to poor performance metric and large variation. Having higher percentage leads to overfitting. This is simulated in mldemos for kNN, variation of F1 measure w.r.t train/test ratio along with the standard deviation(±3σ) is plotted in figure 14. We conclude that 66% train/test ratio gives sufficient samples for training as well as testing, hence results good F1 measure with less variation.

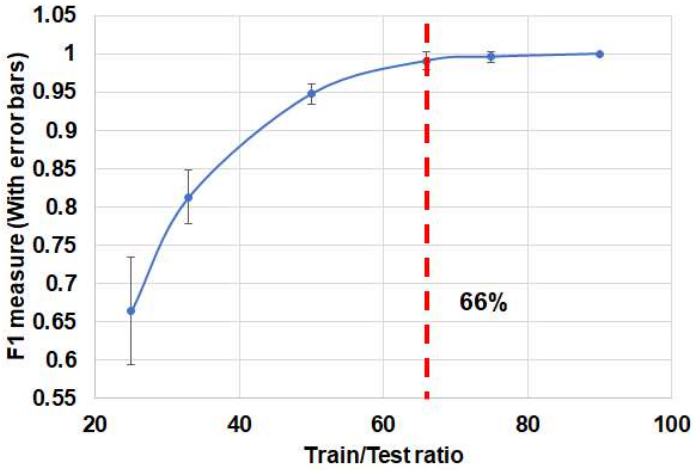


Figure 14: Selection of training/test ratio based on number of points in dataset and F1 measure

## k-Nearst Neighbors (kNN)

From the 100-fold (Selection of 100-fold explained later) cross validation of kNN for various hyperparameter values (k) (shown in figure 15), it is observed that the F1-measure is found to be very good upto 7 nearest neighbours (k=7). We know that the number of exact correct neighbors available in the data set are ~11 since we have chosen train/test ratio to be 66%. Hence, high number of k-will result in poor classification, it is evident from the simulation shown below(K=10, 15). Though performance is good with k=3, outliers can easily influence the classification performance. Hence, K=5 is chosen for the given data set. Euclidean distance is used in all cases.

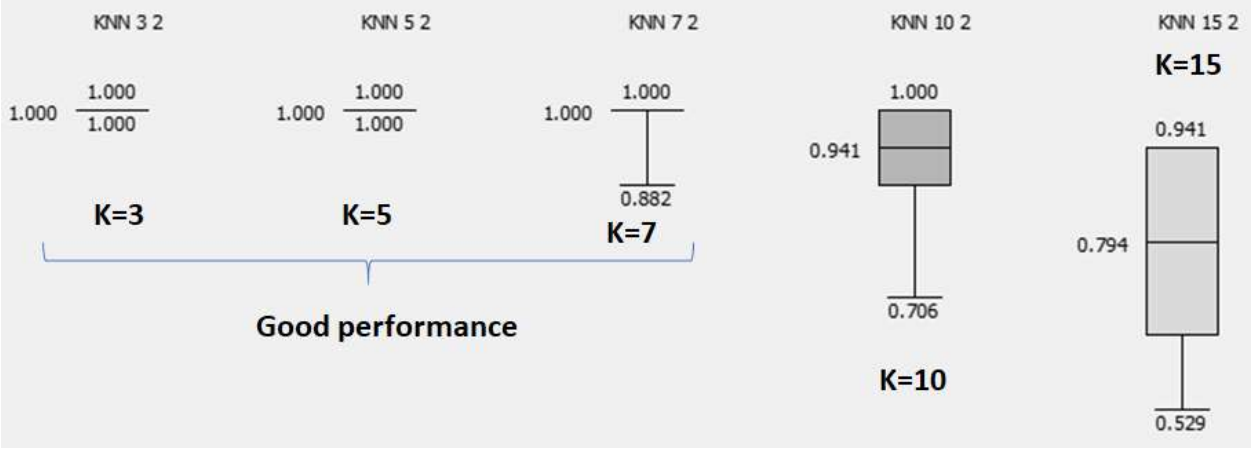


Figure 15: kNN hyperparameters sélection (K=5)

The cross validation is repeated for 5,10,20, 50,100,200 folds for the final selected hyper parameters i.e **k=5, L2 norm, each class with 66%** train to test sample. Standard deviation of the F1 measure for each n-fold simulation (15 cases) is plotted below (left). Based on the results, 100-fold cross validation is selected for validation since there is no further variation in F1 measure beyond 100 folds.

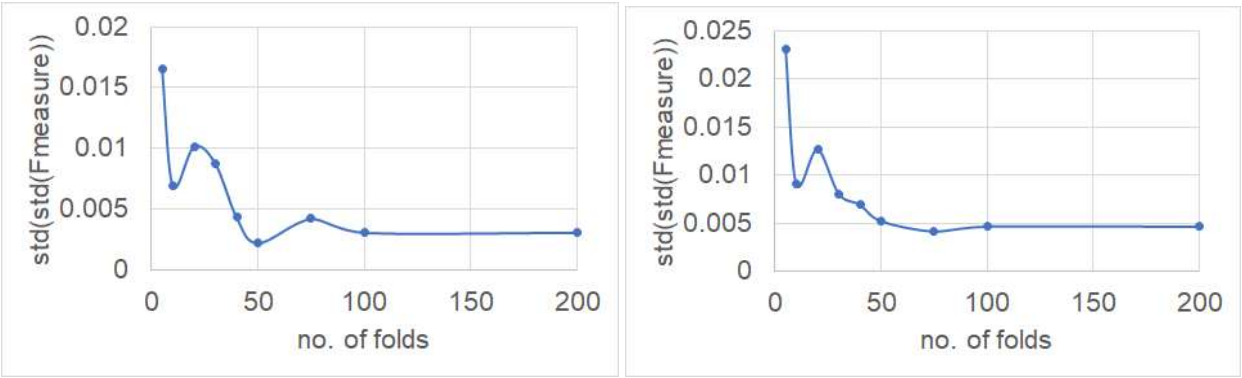


Figure 16: Standard deviation of F1 measure across each of the n-fold comparison(left:KNN, right:GMM)

## Gaussian Mixture Models + Bayes classifier (GMM)

Comparison of GMM for various hyperparameters is shown in figure below it is observed that the F1-measure is found to be insensitive to initialisation method (K-means, random and uniform) for this particular data set. In addition, it is observed that full covariance matrix gives best F1 measure because the classes are correlated with each other, hence this is as expected. Further, the number of components per class = 1 is found to be best choice. This is due to very low number of samples in case of 2 or more components per class in the dataset. It is also observed that train to test sample ratio of 75% gives only marginal improvement in F1-measure(test). Hence, it is decided to use 66% as train to test ratio so that we have at least 5 samples to test the classifier performance.

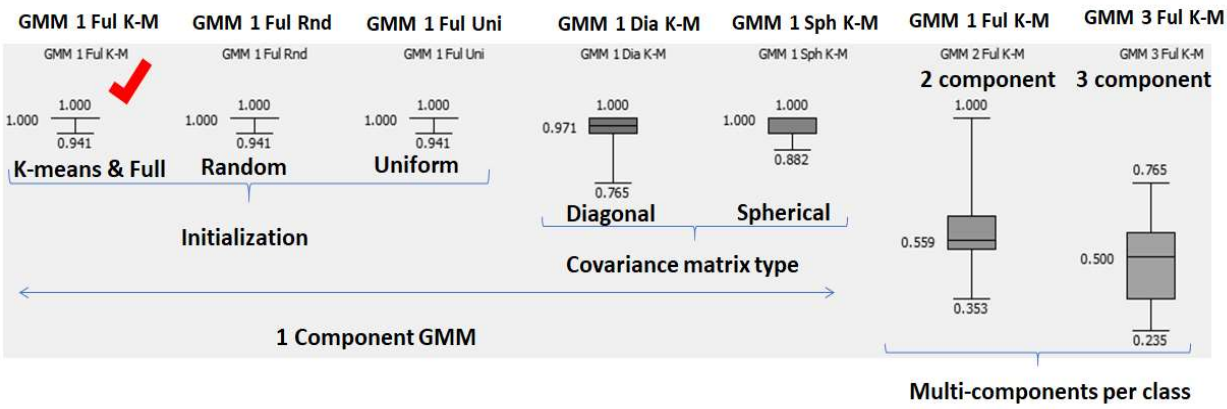


Figure 17: Selection og GMM classifier hyperparameters (Kmeans initialisation, Full covariance)

The cross validation is repeated 15 times for each of 5,10,20,30,40,50,75,100,200 folds for the final selected parameters, i.e **K-means initialisation, Full covariance matrix, one component gaussian for each class with 66% train to test sample**. It is observed that for number of folds > 50, the variation in F1 measure is stable (Figure 16). Hence, 100-fold cross validation is used.

## Support Vector Machine (SVM)

Comparison of SVM is done for various penalty parameters (C=1 to 50) and kernels (Linear, Polynomial, RBF w=0.01, w=0.1). Figure 18(left) shows the F1 measure of 100-fold cross validation for various hyperparameters. It is observed that the F1-measure is very good (=1) upto a penalty parameter value of 10, beyond which the F1 measure deviates. We notice that higher penalty decreases the tolerance too much. We note that the F measure for all the kernels is very close to 1. Hence, a closer observation of the decision boundaries is needed. Figure 18(right) shows the decision boundaries for various kernels (right). We select RBF kernel with penalty factor of 5, since the F1 measure is close to 1 and results in better class separation than the linear kernel. The comparison is performed with train/test ratio of 66%.

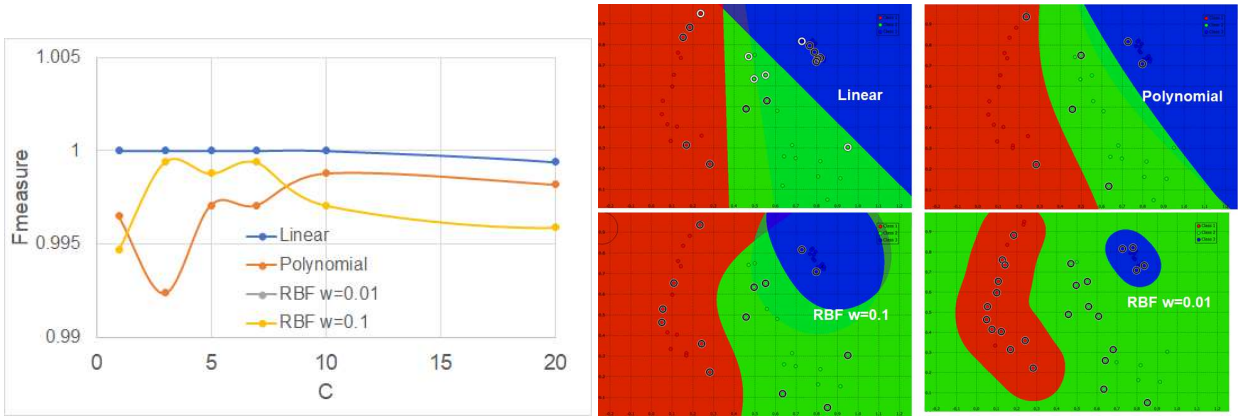


Figure 18: F measurement of different kernel regarding the penalty parameter

## Conclusion

Based on the final selection of hyperparameters across various classification algorithms, 100-fold cross validation is performed to compare the final set of algorithms, the final F1 measure is given in table below. (Train/test ratio = 66%).

| | F-measure (Test) | F-measure(Train) |
|---|---|---|
| **KNN** 5 2 | 0.9935 | 0.9997 |
| **GMM** 1 Ful K-M | 0.9976 | 1 |
| **C-SVM** 5 Lin | **1** | **1** |

# VI. Overall discussion and conclusion

In this project we have been able to appreciate the quality of a projection to make good separable cluster. We selected first five projections based on the required features and percentage of variance explanation by each projection.

We have also have been able to observe different clustering algorithms :K-means algorithms which dependent of a priori knowledge about the number of clusters and the initialization of the centroids of the clusters but the convergence is relatively fast and guaranteed; DBSCAN which is more expensive in terms of computation and has more parameters to tune but can generate non-globular clusters and is robust to outliers if there are noisy measurements.

Classification of the dataset is carried out using kNN, GMM and C-SVM. We selected SVM as the best algorithm for classification. SVM gives best F1 measure on training as well as testing set. Figure 19 shows the performance of classification algorithms along with decision boundaries. Though kNN, GMM and SVM classify the train and test data sets perfectly, we select SVM as best because it gives best F1 measure with no points misclassified and lowest uncertainty in decision boundaries.

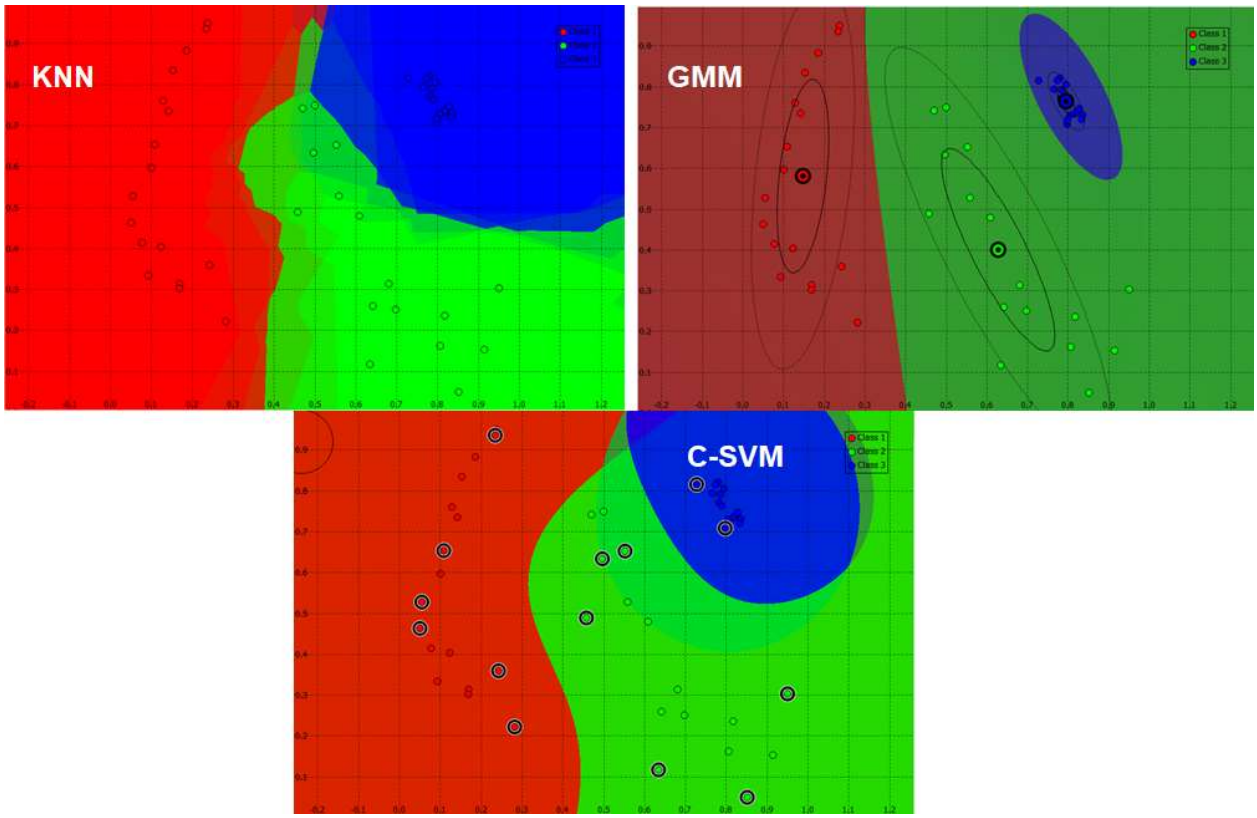


Figure 19: Comparison of decision boundaries for classification algorithms