

Assignment 3:
Unsupervised learning with PCA, t-SNE, k-means,
AHC and SOM
Neuronal and evolutionary computing
2023 - 2024

Pawel Puzdrowski

2023-12-17

Part 1

Dataset1 - provided from course site

No changes to column values were made to the first dataset, only z-score normalization was done on the columns except the last one (class column) with simple jupyter notebook "dataCleaningNotebooks.ipynb" which is situated in directory "DataCleaning". The file for this dataset is in directory "datasets" and the name is "A3-dataset1Modified.csv".

Dataset from internet - Dry bean dataset

The second dataset was obtained from the kaggle.com website specifically [1]. This website consists of 16 variables, one class column and 2501 features. The database represents data that was obtained from a computer vision system where seven different dry beans were examined by a high resolution camera and over 13 000 images were taken. The website provides two csv files, one for testing and one for training, only the training csv file is used because the criterium for this task that the dataset needs to have at least 200 features and the training csv file has 2500 features which is enough.

This dataset class column values are categorical values (text) which represents different beans. There exist seven different beans names and they were changed for numbers. How the bean names were changes in the class column is shown in table 1. Z-score normalization was applied to every column because the columns had a lot of outliers, the last column, class, was not normalized. The modification and normalization was done with script "dataCleaningNotebooks.ipynb" which is in directory "DataCleaning".

Old value	New value
SEKER	0
BARBUNYA	1
BOMBAY	2
CALI	3
DERMASON	4
HOROZ	5
SIRA	6

Table 1: Bean names change

Part 2

PCA

Implementation of PCA is situated in directory "PCA" where a jupyter notebook file exist, "PCA.ipynb".

Scatter plot and scree plot for dataset 1 is visualized in figure 1 and 2 respectively. Plot for dataset2 is visualized in figure 3 and 4.

The PCA doesn't do a good job in classifying the different classes in dataset 1. The scatter plot is spread over the graph and it's hard to see the clusters. Scatter plot for dataset 2 on the other hand is much better and the clusters are visible.

Dataset1

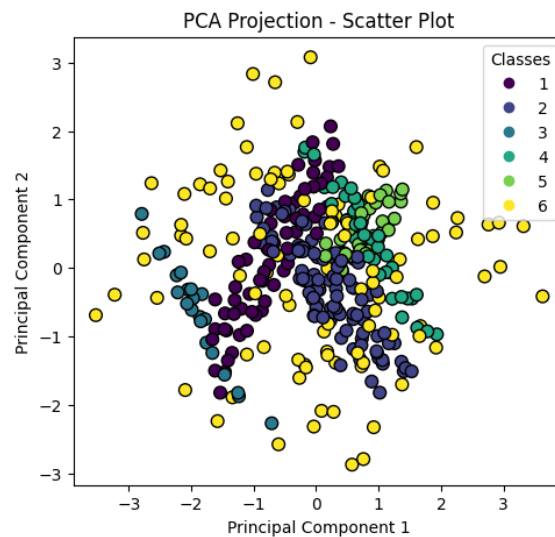


Figure 1: Dataset1 scatter plot

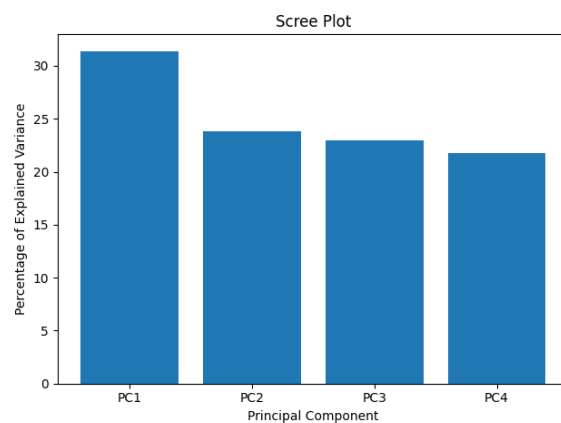


Figure 2: Dataset1 scree plot

Dataset2

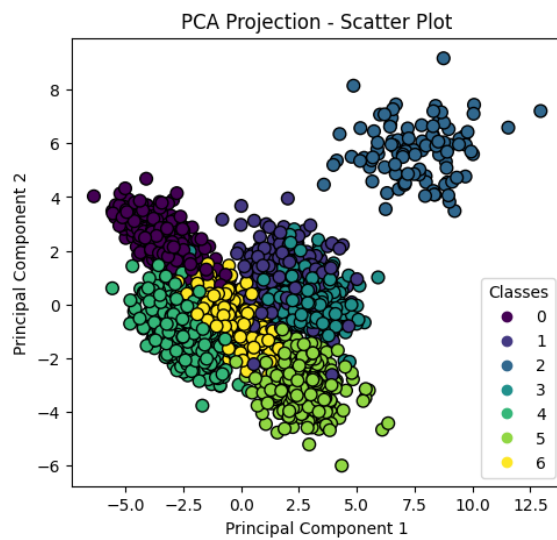


Figure 3: Dataset2 scatter plot

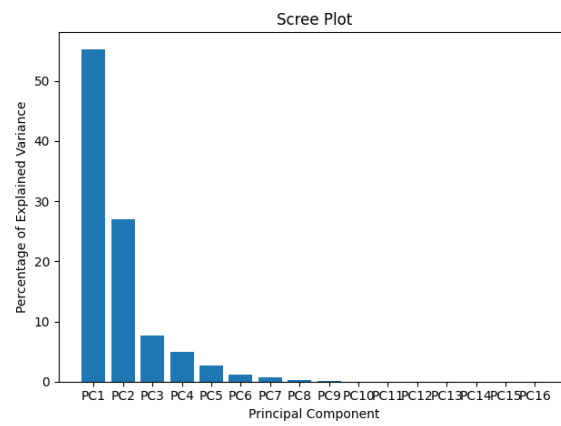


Figure 4: Dataset2 scree plot

t-SNE

Documentation for t-SNE in python was accessed by the official scikit learn website [2]. According to the documentation the important parameters that can be tuned are following:

- perplexity - values should be between 5 and 50. Is connected to number of nearest neighbours, for bigger datasets this parameter should be bigger.
- learning_rate - How fast the model classifies the data, the value for this parameter is given by: $\frac{\text{Number of samples}}{\text{early_exaggeration} \cdot 4}$
- n_iter - number of iteration

Other parameters in t-SNE have either value "None" or have no impact on the classification if the parameter changes. These are the other parameters:

- early_exaggeration - value: the program changes this parameter automatically. Defines how much space will be between clusters
- n_components - value: default=2
sets the dimension of embedded space, the task description says that the t-SNE projection should be found in 2D.
- n_iter_without_progress - value: default=300
Maximum number of iterations without progress. Decreasing and increasing the value doesn't affect the output plot.
- min_grad_norm - value: default=1e-7
Decreasing and increasing the value doesn't affect the output plot.
- metric - default='euclidean'
Method to calculate the distance between points, other methods have no change in the output plot.
- metric_params - value: default=None
Additional keyword arguments for the metric function. Hard to find other specification for this parameter
- init - value: default="pca"
Other value for this parameter is "random" and it spreads out the clusters and changes the value so the output plot is not better.

- verbose - value: 1
If set to 1, gives more information what happens at every step of the iterations.
- random_state - value: default=None
Randomize the data, doesn't have an effect on the output plot.
- method - value: default='barnes_hut'
Gradient calculation algorithm. The other option for this parameter is "exact", the program is slower and doesn't show any improvement in the output plot.
- angle - value: default=0.5
According to the documentation, this variable doesn't have big effect on the output.
- n_jobs - value: default=None
Parallelization of the program. Not relevant for this assignment.

Dataset 1

Dataset1	Test 1	Test 2	Test 3	Test 4	Test 5	optimal values
Perplexity	20	5	5	5	20	5
Learning_rate	2	30	2	5	30	2
n_iter	1000	1000	1500	500	1000	1000

Table 2: Dataset 1 parameter values

The different values for t-SNE parameters are shown in table 2, five columns are for test values and one column for optimal values for dataset 1. The output plot from this tests are visualized in the following figures:

- Increased perplexity - figure 5b
- Increased learning rate - figure 5a
- Increased and decreased number of iterations - figure 6
- Increased both learning rate and perplexity - figure 7

The output plot from the optimal parameter values is shown in figure 8. Perplexity was chosen to be the lowest value 5, because the dataset 1 is a small database and according to documentation the value for this parameter should be small with small datasets. Learning rate value is obtain by the following equation:

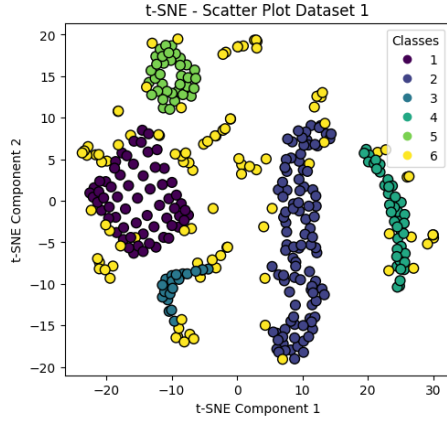
$$\frac{\text{Number of samples}}{\text{early_exaggeration}} \frac{1}{4}$$

early_exaggeration is chosen automatically by the program and the value was 61 and the final learning rate was calculated to be:

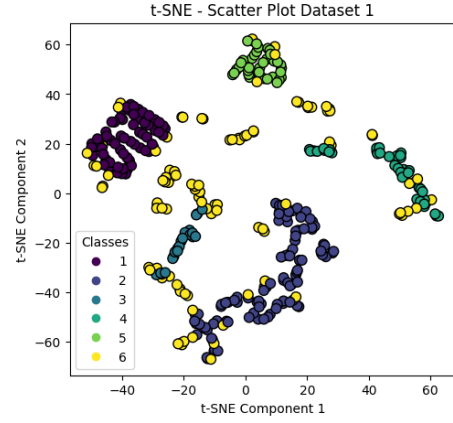
$$\frac{\frac{361}{61}}{4} \approx 2$$

Number of iteration was set to 1000, increasing or decreasing it more didn't give better output, according to figure 6.

The t-SNE program seems to do a pretty good job in clustering almost all data points except the class 6 datapoints. Wasn't able to find parameter values that could handle all classes including the sixth one.

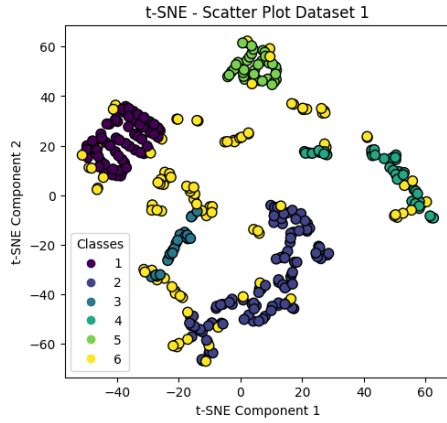


(a) Increased learning rate

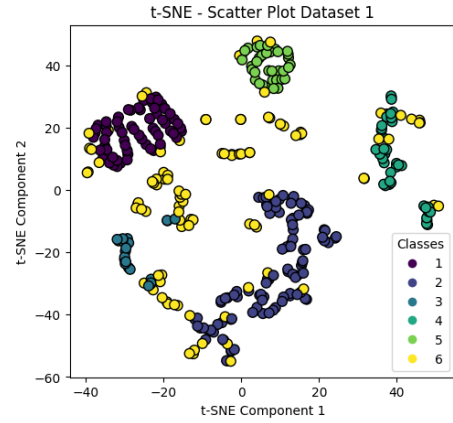


(b) Increased perplexity

Figure 5: Output plot for increased learning rate and perplexity for dataset 1



(a) Increased number of iterations



(b) Decreased number of iterations

Figure 6: Increased and decreased number of iterations output plot

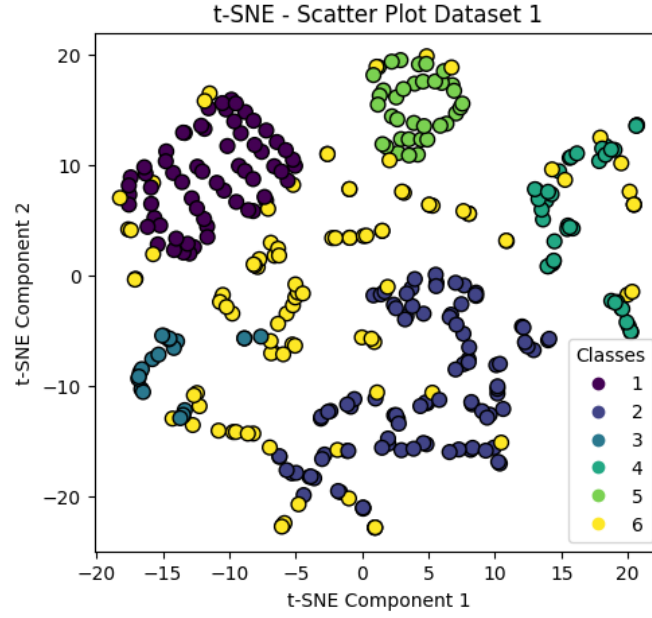


Figure 7: Dataset 1 increased learning rate and perplexity

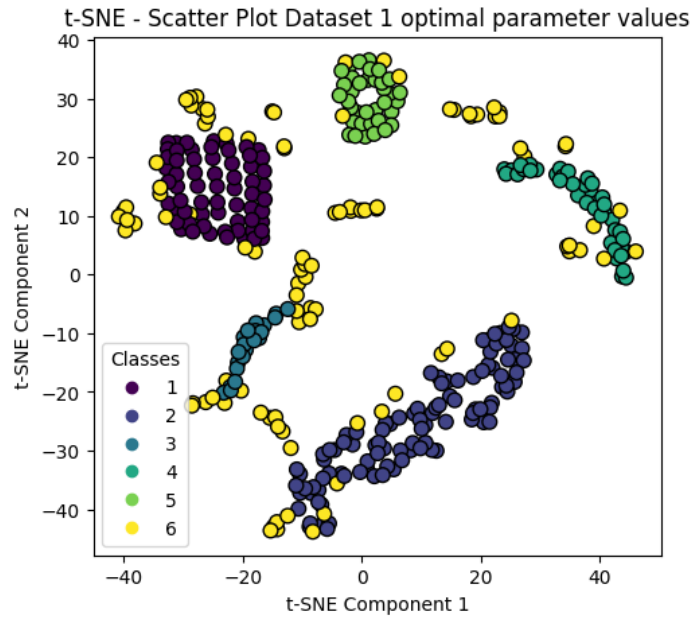


Figure 8: Dataset 1 optimal parameter values plot

Dataset 2 - dry beans

The different parameter values that were tested are shown in table 3.

All the parameter values gave almost the same output plot of the different classes, the best one was the one with perplexity 30, learning rate 10 and number of iterations 1000, which is shown in figure 9 but can even say that the perplexity: 50 was good as well. The worst was the one with perplexity 20 because the clusters are more spread.

As shown in figure 10, number of iteration has a slightly influence on the clustering. Figure 11 shows different tests with different values for perplexity and for this dataset higher value for perplexity seems to be an advantage. Learning rate on the other hand, as shown in figure 12, doesn't have big effect on the t-SNE output.

The optimal values where obtained as follows:

Learning rate, as in the previous dataset, was obtained by this equation:

$$\frac{\frac{\text{Number of samples}}{\text{early-exaggeration}}}{4}$$

The value for optimal learning was calculated to be:

$$\frac{\frac{2500}{68}}{4} \approx 10$$

The perplexity was set to be higher than for the dataset 1 because dataset 2 is larger and perplexity should be larger when database increases. Number of iterations was set to 1000 because increasing more didn't give any better output plot.

Dataset1	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Optimal values
Perplexity	20	35	50	30	30	30	30	30
Learning_rate	10	10	10	30	100	10	10	10
n_iter	1000	1000	1000	1000	1000	500	1500	1000

Table 3: Dataset 2 parameter values

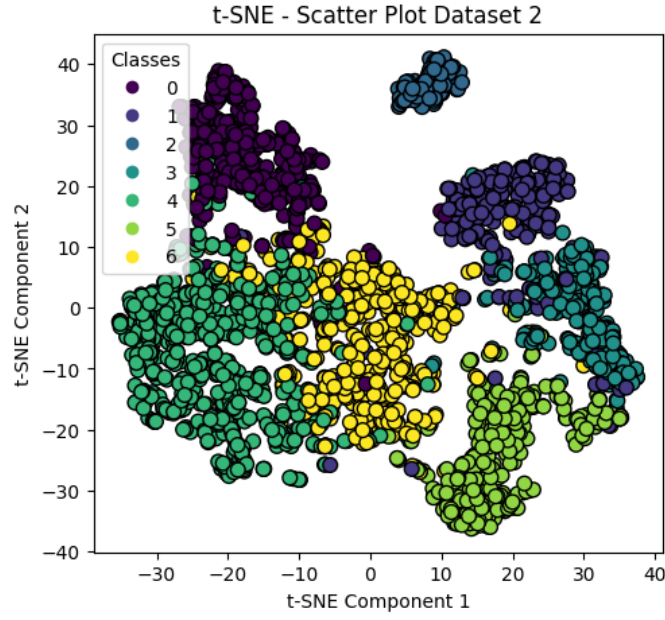
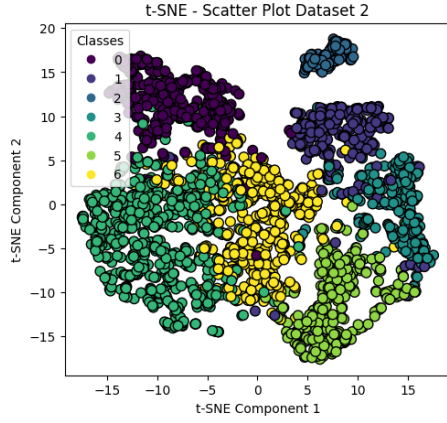
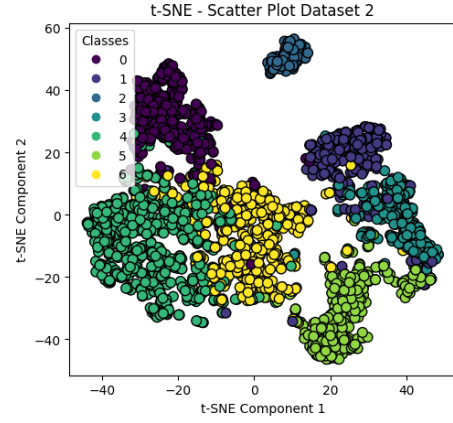


Figure 9: Dataset 2 optimal values: perplexity: 30, learningRate: 10, nIter: 1000

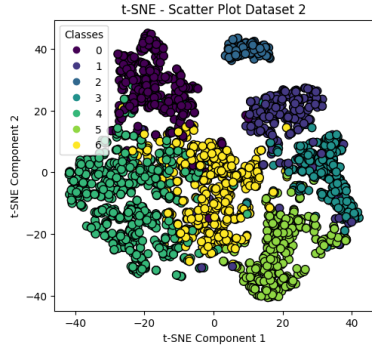


(a) Number of iteration: 500

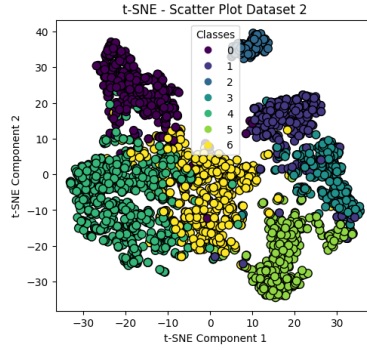


(b) Number of iteration: 1500

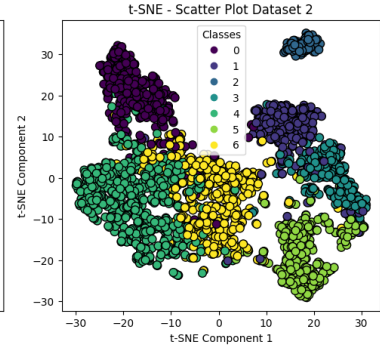
Figure 10: Database 2: Different number of iterations with perplexity: 30, learningRate: 10



(a) Perplexity = 20

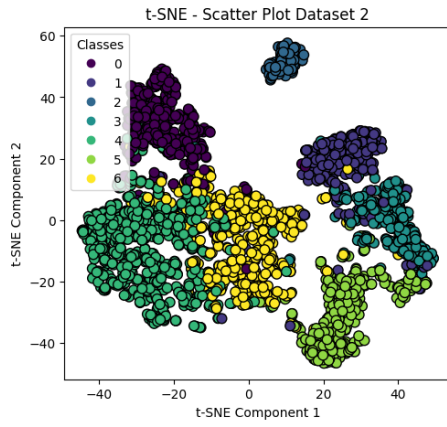


(b) Perplexity: 35

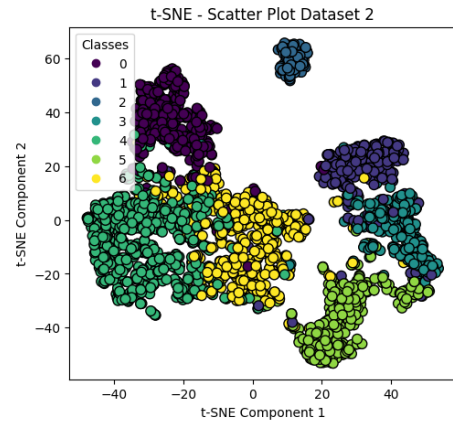


(c) Perplexity: 50

Figure 11: Database 2: Different perplexities, learningRate: 10, nr_iter = 1000



(a) learningRate: 30



(b) learningRate: 100

Figure 12: Database 2: Different learningRate, perplexity:30, nr_iter = 1000

k-means

AHC

SOM

Bibliography

- [1] Dry bean classification. <https://www.kaggle.com/datasets/gauravduttakiit/dry-bean-classification>. Accessed 2023-12-08.
- [2] sklearn.manifold.tsne. <https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html>. Accessed 2023-12-11.