Foundations of Data Analysis
Lab Exercise no.2 – Unsupervised Learning

Student: Jovana Krtolica - Matriculation Number: a11946909
University of Vienna, Faculty of Computer Science

The submission for the laboratory exercise no.2 includes five Python files. The files are named based on the tasks from the assignment sheet such that the file **exercise1.1.py** is for the task 1 subtask a), the file ***exercise1.2.py*** is for the task 1 subtask b), the file ***exercise1.3.py*** is for the task 1 subtask c), the file ***exercise2.1.py*** is for the task 2 subtask a), the file ***exercise2.2-2.3.***py is for the task 2 with subtasks b) and c) and the file ***exercise3.1.py*** is for task 3 with subtask a). In order to compile and run the tasks, the user will have to open the Python files in PyCharmIDE, right click on the file and click on the 'Run' button. It is important that the required data sets are also included in the same directory as the Python files.

As external materials and resources, I have used the slides and video materials posted on the FDA Moodle course page, the recommended literature, the official Sklearn documentation, the official Numpy documentation and the tutorials with the links below. In the code, the resources are also cited in order to know which link is used for which task and subtask. The links to the external resources from the internet are the following:

https://machinelearningmastery.com/calculate-principal-component-analysis-scratch-python/
https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60
https://towardsdatascience.com/principal-component-analysis-pca-from-scratch-in-python-7f3e2a540c51
https://stackoverflow.com/questions/32857029/python-scikit-learn-pca-explained-variance-ratio-cutoff
https://stackoverflow.com/questions/57293716/sklearn-pca-explained-variance-and-explained-variance-ratio-difference
https://vitalflux.com/pca-explained-variance-concept-python-example/
https://scikit-learn.org/stable/auto_examples/cluster/plot_dbscan.html

Task explanations:

1) **Exercise1.1:**

In this subtask, I am first loading the seeds.csv file. Since I am using only native Python libraries and numpy, the next step is to scale the data by calculating the mean of the data and then centering the data by doing a transpose on the division between the seeds data and the mean of the seeds. After the scaling step, the covariance matrix is calculated from which the eigendecomposition process happens. The eigendecomposition process decomposes the matrix into eigenvalues and eigenvectors. The eigenvalues are sorted in descending order along with their corresponding eigenvectors. Then I select the principal components and make a subset from the rearranged eigenvalues. Finally, the data is projected through transformation by doing a dot product between the eigenvalue subset transpose and the centered mean transpose.

2) **Exercise 1.2:**

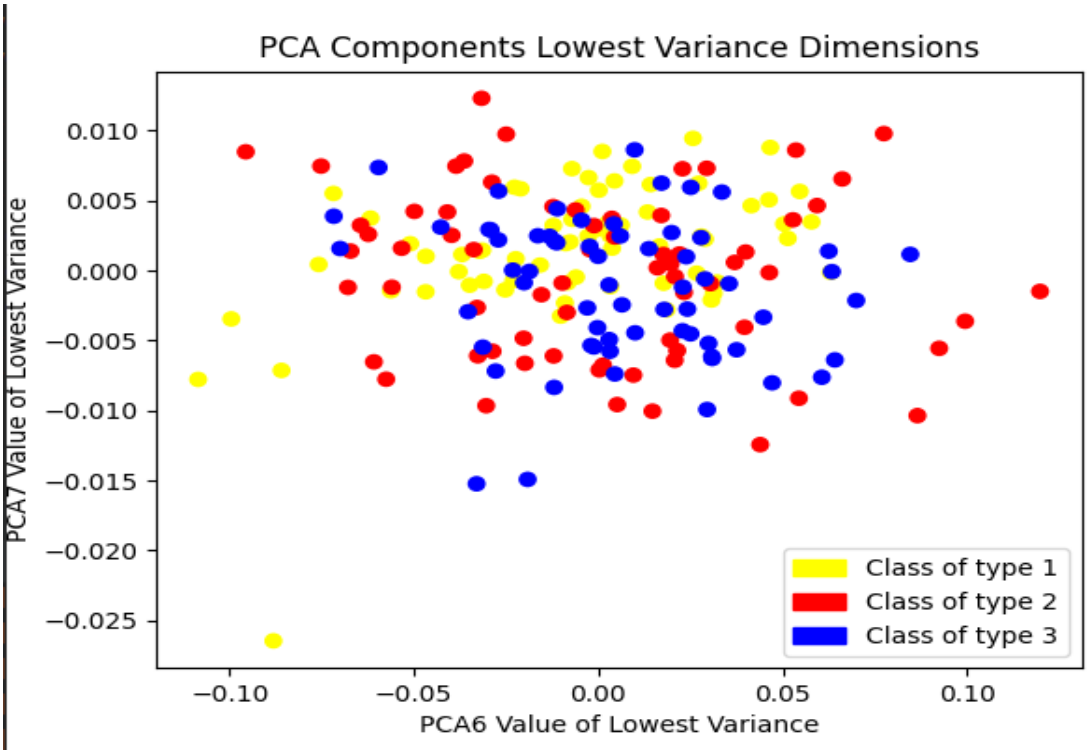
In this subtask, I am repeating the manual steps for implementation of the PCA through native Python libraries. After the manual implementation from scratch is finish, I am calculating the explained variance and the explained variance ratio through the sorted eigenvalues. For the comparison of the results, I am using the built in PCA functionalities from Sklearn. I am doing a decomposition and fit transform the seeds data without their type. Then I calculate the explained variance and the explained variance ratio, where I get the following results.

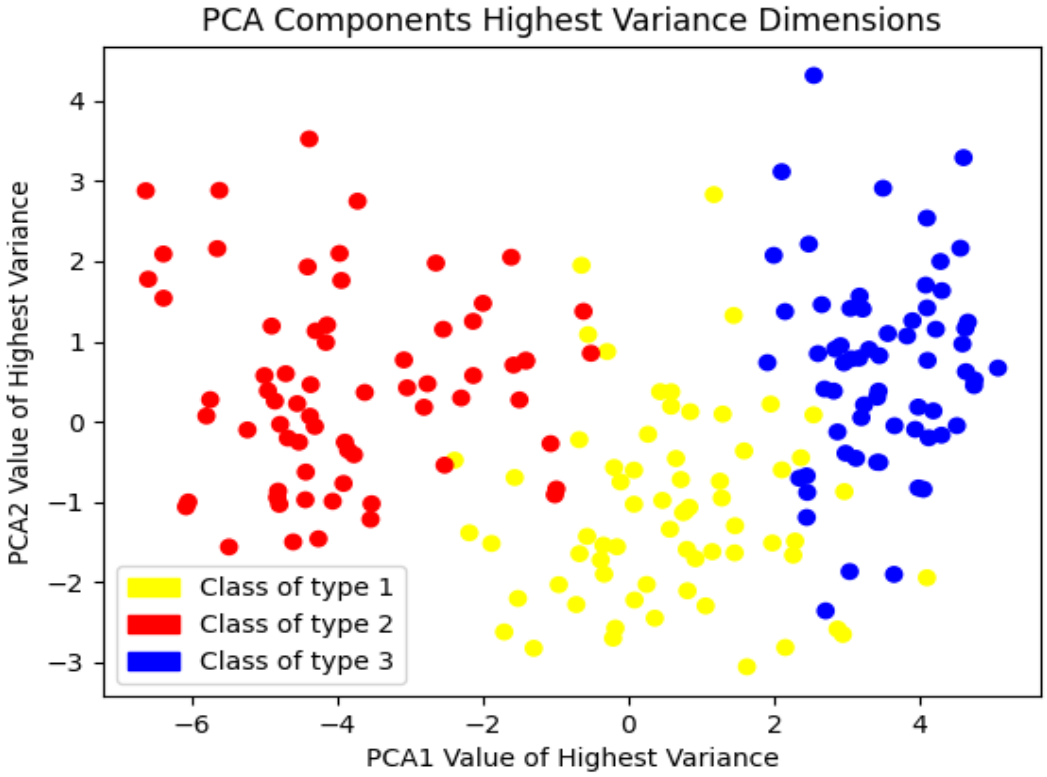
```
EXPLAINED WITH NATIVE LIBRARIES
[1.08516254e+01 2.04834826e+00 7.37914840e-02 1.27479228e-02
 2.75600772e-03 1.58425236e-03 2.91698076e-05]
[8.35326268e-01 1.57675836e-01 5.68025183e-03 9.81297676e-04
 2.12149385e-04 1.21951096e-04 2.24540616e-06]
EXPLAINED WITH PCA
[1.08516254e+01 2.04834826e+00 7.37914840e-02 1.27479228e-02
 2.75600772e-03 1.58425236e-03 2.91698076e-05]
[8.35326268e-01 1.57675836e-01 5.68025183e-03 9.81297676e-04
 2.12149385e-04 1.21951096e-04 2.24540616e-06]
COMPARISON RESULTS:
1.0
1.0000000000000002
```

3) **Exercise 1.3:**

In this subtask, I am repeating the steps from the subtask 1.2 where I just add the plots. Based on the three types, I am coloring the data in blue, red and yellow. I am plotting the

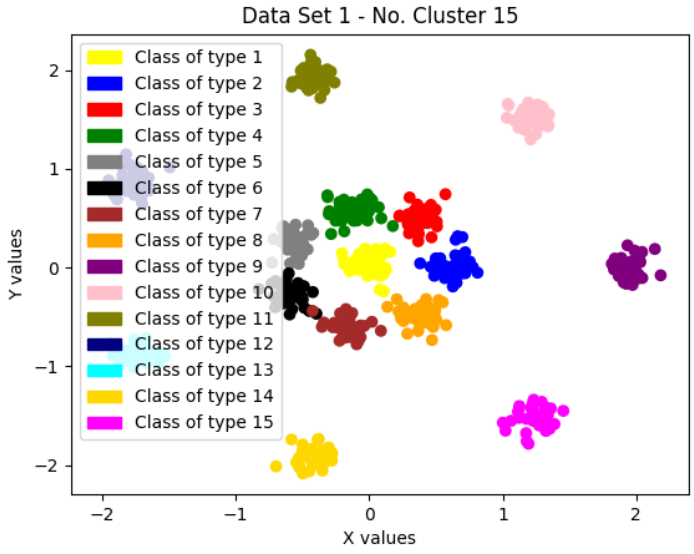
two PCA with the highest variance and the two PCA with the lowest variance. The results from the plots are the following based on their Type labels:

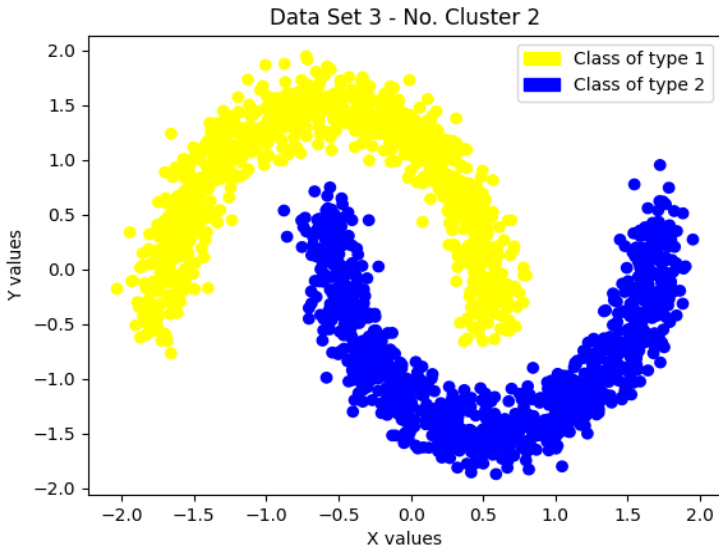


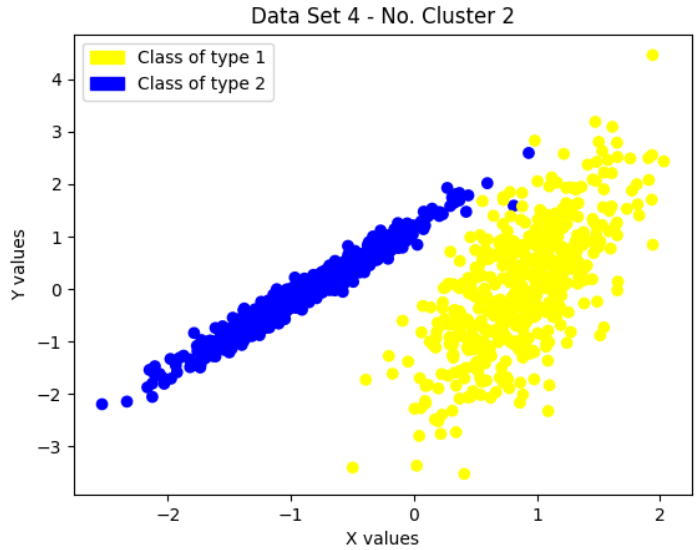


4) Exercise 2.1:

In this subtask, the first thing I do is to import the four different data sets. Then I am deleting the Type label. Before deleting them, I extract the type in order to use it later for the coloring. I am processing the data by using the StandardScalar from Sklearn and the fit transform method on all the four datasets. Then I am coloring the data based on their type labels. For the first data set there are 15 different types colored in different color, whereas for the three other datasets only two types exist which are also colored in different color. Then I am plotting the data based on their type labels with different colors and the legend shows the classes and their colors, which can be seen on the pictures below for each of the datasets.







5) **Exercise 2.2-2.3:**

In this file, I am doing the subtask b) and subtask c) from the task number 2.

Part b:

First, I am doing the reading and importing of the four data sets. I extract the type labels from each of the datasets and then delete the type column from the datasets. The data processing is done with the StandardScaler from Sklearn. Then I am computing four different clustering techniques for each of the datasets. The techniques which I have used are the DBSCAN, KMeans algorithm, Expectation Maximization algorithm and the Average Link algorithm. These algorithms are imported from built in features from Sklearn and for each algorithm for each dataset I am extracting the labels. For the DBSCAN I have chosen epsilon values which were giving the best results based on the plot which I have plotted for each of the datasets and the minimum samples is four because with this number I got the best results. The epsilon values are different for each dataset. In the KMeans algorithm, the number of clusters is fifteen for the first data set because there are fifteen different types and two for the three other datasets because there are only two different types. The random state variable makes the results reproducible. For the Expectation Maximization algorithm, I am using the Gaussian Mixture and also the number of components is the same as in the KMeans algorithm, fifteen for the first dataset and two for the other three datasets. For the Average Link algorithm, I am using the agglomerative clustering and the number of clusters is equal to fifteen for the first dataset and two for the three other datasets. The plots will be shown in the subtask d) where I will discuss the results.

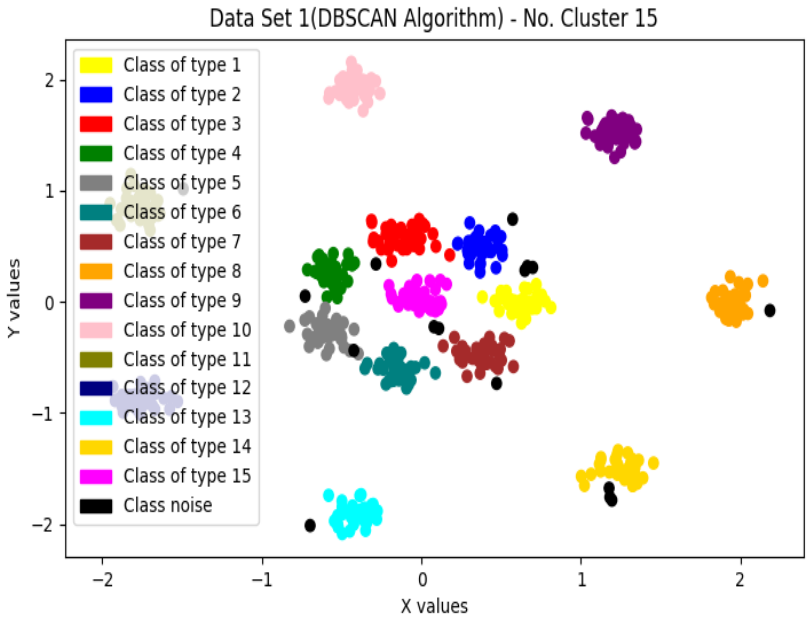
Part c:

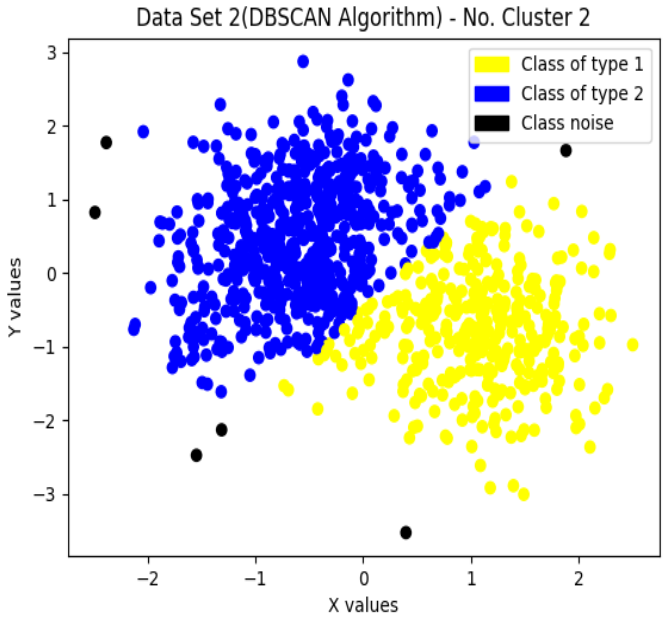
After the part b) of the second exercise is finished, I have computed the normalized mutual information for each algorithm in each dataset and then found the adjusted rand index. These two parameters were calculated by using Sklearn and by using the type labels for the particular dataset and the labels received from each of the algorithms based on the dataset. Based on the received results, the evaluation of the clustering was conducted, where I have debugged and adapted the values in order to get more precise and closer results. The results shown in the subtask d).

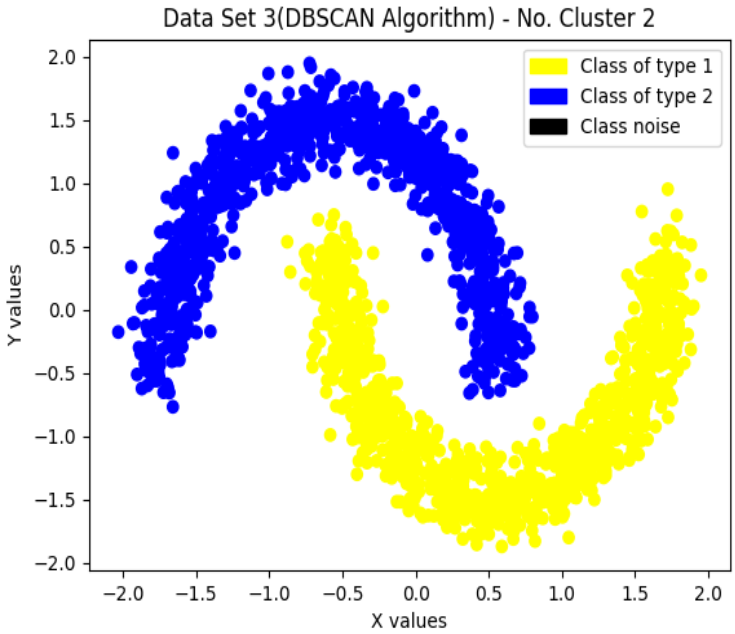
6) **Exercise 2d – Discuss your results:**

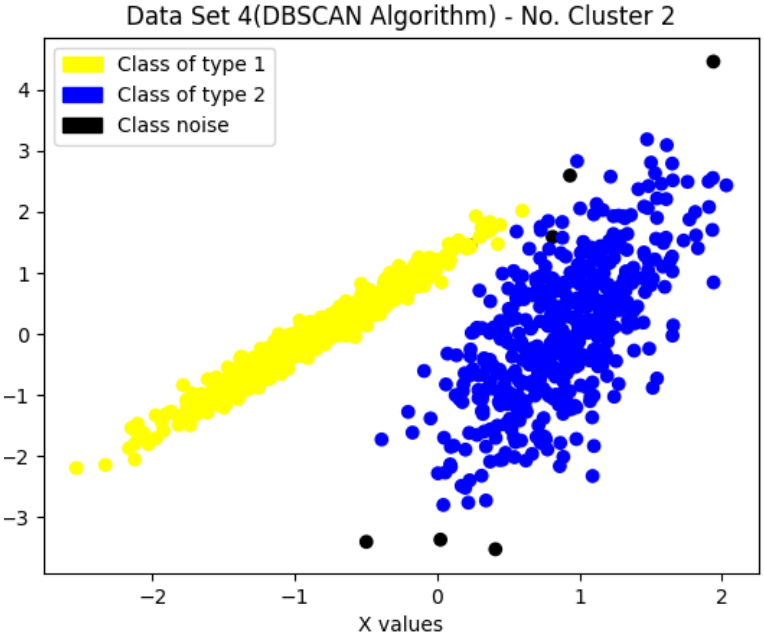
Based on part 2b and 2c, the results can be evaluated and viewed on the plots. For the evaluation, the most interesting is the DBSCAN technique where the noise is shown and colored with black. The noise means that for that data the algorithm can not determine

to which class label it belongs. The results for the DBSCAN algorithm are shown in the following pictures. For the dataset number 3, there is no noise as the clustering is perfectly made. For all the other datasets, the black dots represent the data which was not clustered.





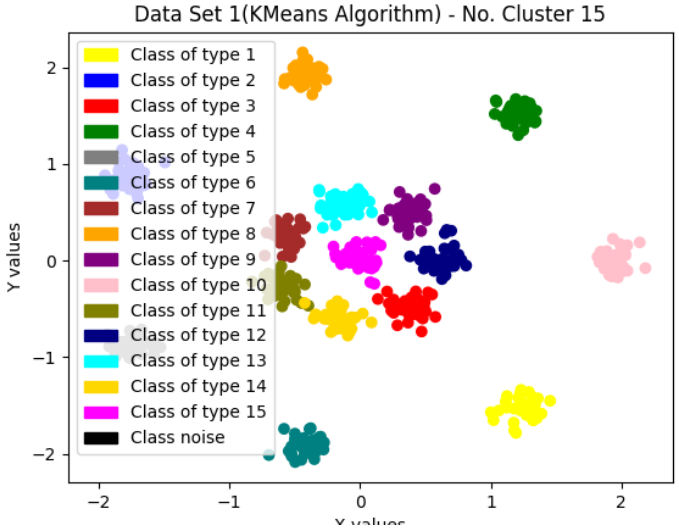


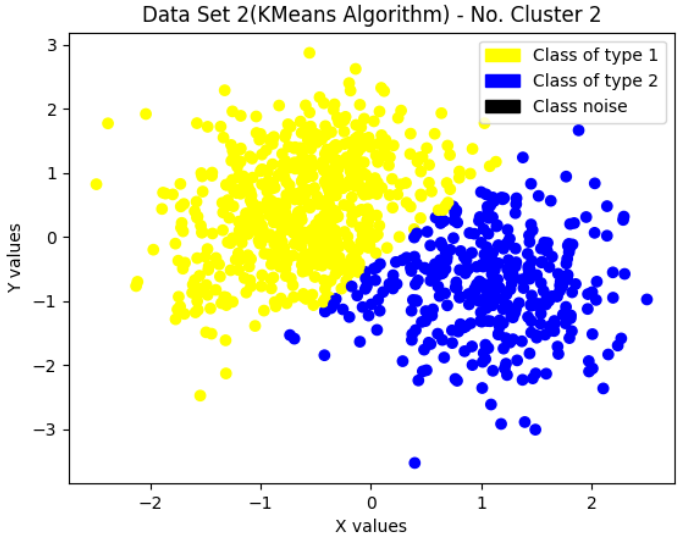


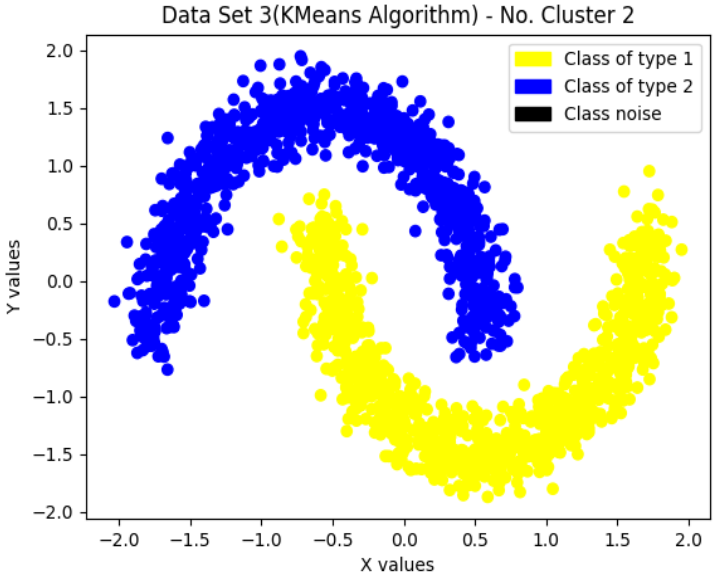
Based on the results received for 2b, we can see that the rand index and the NMI are almost close together and close to 1, as it can be seen on the following picture:

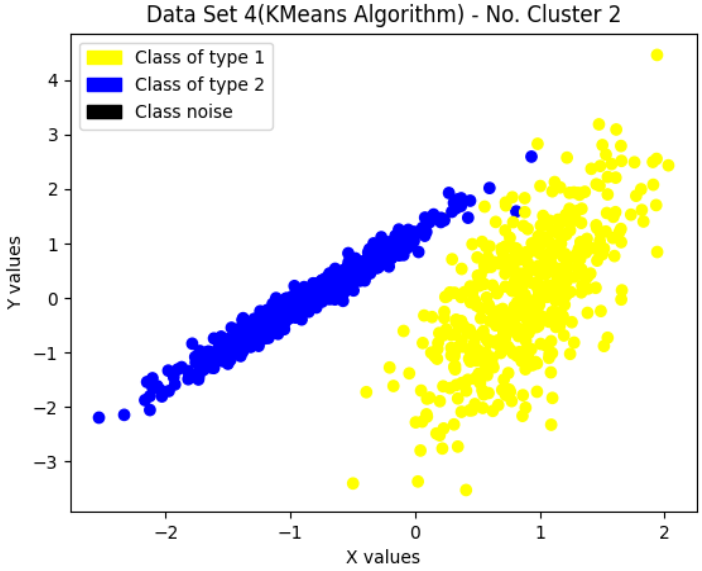
```
Normalized Mutual Information DBSCAN DataSet1:  0.9685832479037582
Adjusted Rand Score DBSCAN Dataset1:  0.966284556175396
Normalized Mutual Information DBSCAN DataSet2:  0.9699581497412587
Adjusted Rand Score DBSCAN Dataset2:  0.9869190348751878
Normalized Mutual Information DBSCAN DataSet3:  1.0
Adjusted Rand Score DBSCAN Dataset3:  1.0
Normalized Mutual Information DBSCAN DataSet4: 0.9682240410382984
Adjusted Rand Score DBSCAN Dataset4:  0.9860738679914162
```

For the KMeans algorithm, there is no noise shown as it can be seen on the plots and the results for the NMI and the Rand Index are 1.0 for the dataset2, dataset3 and dataset4 whereas for dataset1 it is close to 1.0. This is shown on the plots and the results shown in the following pictures:



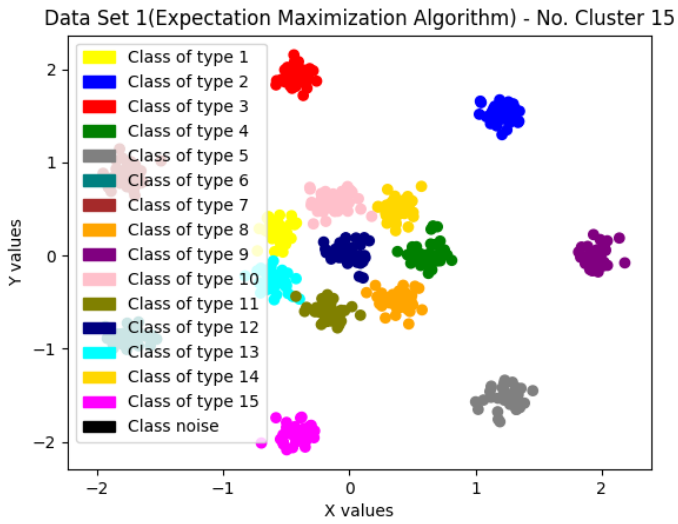


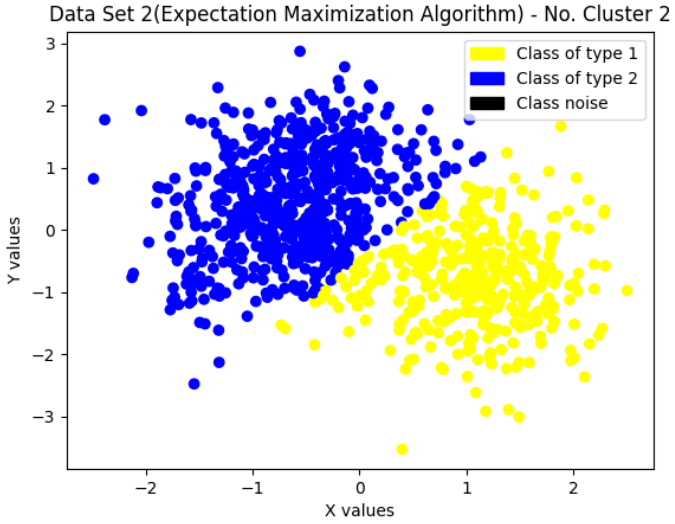


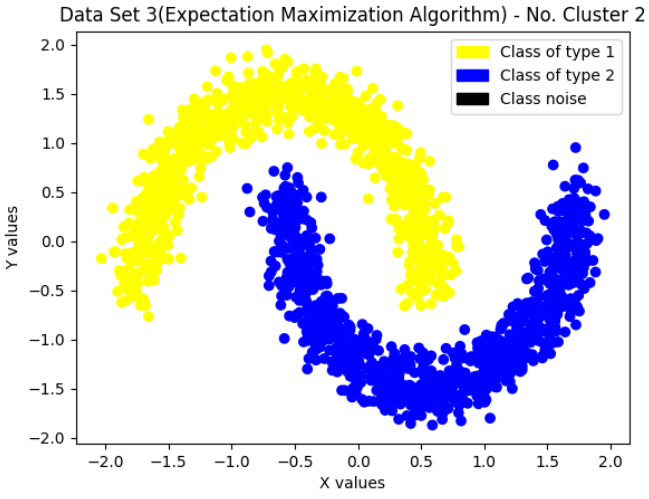


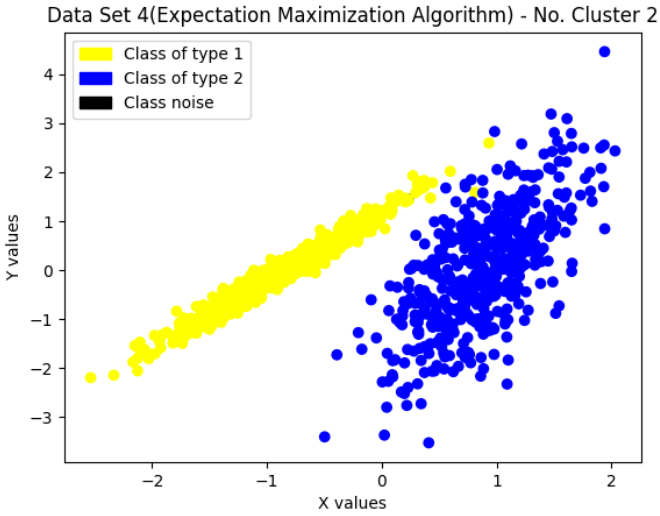
```
Normalized Mutual Information KMeans DataSet1:  0.9971142759580855
Adjusted Rand Score KMeans Dataset1:  0.9963889561472716
Normalized Mutual Information KMeans DataSet2:  1.0
Adjusted Rand Score KMeans Dataset2:  1.0
Normalized Mutual Information KMeans DataSet3:  1.0
Adjusted Rand Score KMeans Dataset3:  1.0
Normalized Mutual Information KMeans DataSet4:  1.0
Adjusted Rand Score KMeans Dataset4:  1.0
```

For the Expectation Maximization, there is no noise for any of the datasets and the results for the Rand Index and the NMI are equal to 1.0 for each of the datasets as shown in the picture following:



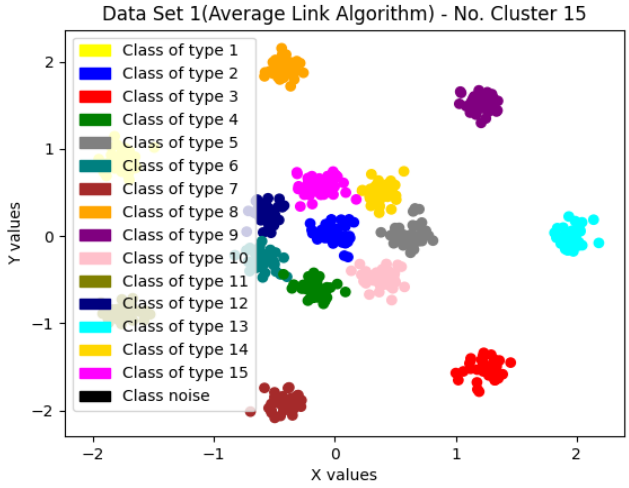


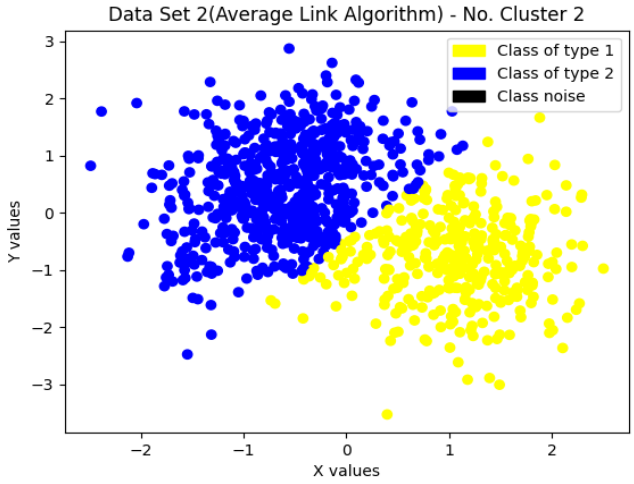


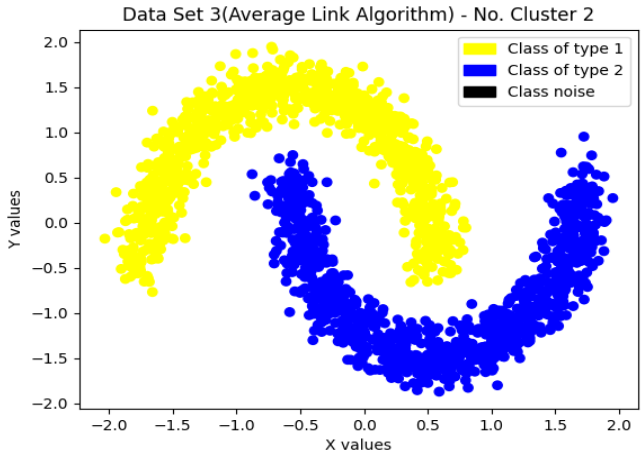


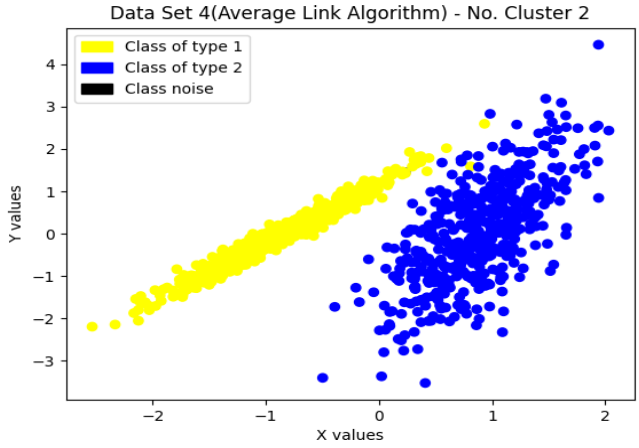
```
Normalized Mutual Information Expectation Maximization DataSet1:  1.0
Adjusted Rand Score Expectation Maximization Dataset1:  1.0
Normalized Mutual Information Expectation Maximization DataSet2:  1.0
Adjusted Rand Score Expectation Maximization Dataset2:  1.0
Normalized Mutual Information Expectation Maximization DataSet3:  1.0
Adjusted Rand Score Expectation Maximization Dataset3:  1.0
Normalized Mutual Information Expectation Maximization DataSet4:  1.0
Adjusted Rand Score Expectation Maximization Dataset4:  1.0
```

For the Expectation Maximization, there is no noise for any of the datasets and the results for the Rand Index and the NMI are equal to 1.0 for each of the datasets as shown in the picture following:










```
Normalized Mutual Information Average Link DataSet1:  1.0
Adjusted Rand Score Expectation Average Link Dataset1:  1.0
Normalized Mutual Information Average Link DataSet2:  1.0
Adjusted Rand Score Expectation Average Link Dataset2:  1.0
Normalized Mutual Information Average Link DataSet3:  1.0
Adjusted Rand Score Expectation Average Link Dataset3:  1.0
Normalized Mutual Information Average Link DataSet4:  1.0
Adjusted Rand Score Expectation Average Link Dataset4:  1.0
```

7) **Exercise 3.1:**

In this subtask, I am first reading the file and then split the file based on lines. I create a dictionary and iterate through the dictionary by splitting the lines by semicolon. Then I check to find the key value pair for frequent songs and add them to a file.