Foreword: A Paragraph of Thanks

I would like to heartfelt thank you all for your effort and patient you put for the assignments so far. I believe each task given in these assignments has enabled you to make a great deal of practices on solving different data mining tasks through different learning algorithms, and therefore, this course can be considered as a cornerstone for understanding the importance of data science.

1 Introduction

In this final assignment, you are going to have an opportunity to gain skills on dealing with unlabelled data by clustering algorithms. In addition, you will get insight on how clustering algorithms are used as a preprocessing. After then, you will also handle with outlier detection problem by applying a clustering algorithm.

1.1 Clustering

Clustering is a process aiming at finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups. It is known as unsupervised learning task in data mining. Therefore, it does not consider the observations of the tuples in data set. That's why, it is an important task in data science when considering the fact that labelling high volume of data is tiring, time consuming, and even prone-to-errors in case of manual labelling.

There are two types of clustering: *partitional* where data objects are divided into non-overlapped clusters, and *hierarchical* where a set of nested clusters organized as a hierarchical tree. While K-means is known as the most popular partitional clustering algorithm; agglomerative and divisive clustering approaches are often used for hierarchical clustering.

K-means is a simple-to-run clustering algorithm. It initially assigns every data objects to the closest clusters that are initially determined in a random manner; then, updates the centroid of the clusters such that it is at the center of the objects that are assigned to. Step-by-step, K-means finds optimal cluster centroids at which no longer convergence can be observed. There are different metrics that can be adopted for analysing the similarity between data objects. K-means possesses two parameters: *i*) number of clusters and *ii*) initial centroid of the clusters. The performance of K-means is highly dependant on these parameters; and hence, these should be properly adjusted. The pseudo-code of K-means clustering algorithm is given in Algorithm 1.

Algorithm 1: General steps of K-Means Clustering

```
Select K points as initial centroids;
repeat
    Form K clusters by assigning all data objects to the closest centroid;
    Recompute the centroid of each cluster;
until The centroids no longer change;
return Cluster centroids
```

1.2 Anomaly Detection

Anomaly, also known as outlier, is a set of data points that are considerably different than the remainder of the data. Unlike to the noise, detection of anomalies in data set is very important as anomalies may lead to a catastrophic outcomes in a real life because they are important factors on misleading the learners.

The objective with the outlier detection is to find all data points in a given data set $x \in D$ with anomaly scores greater than some threshold t. Visual, statistical, distance-based, density-based, and clustering-based approaches are often applied for detection of the outliers.

2 Clustering Task

2.1 Visualization of steps in K-means

In this task, you are expected to demonstrate the running principle of K-means algorithm on a given two dimensional fabricated data set. An example for this demonstration is given in Figure 1.

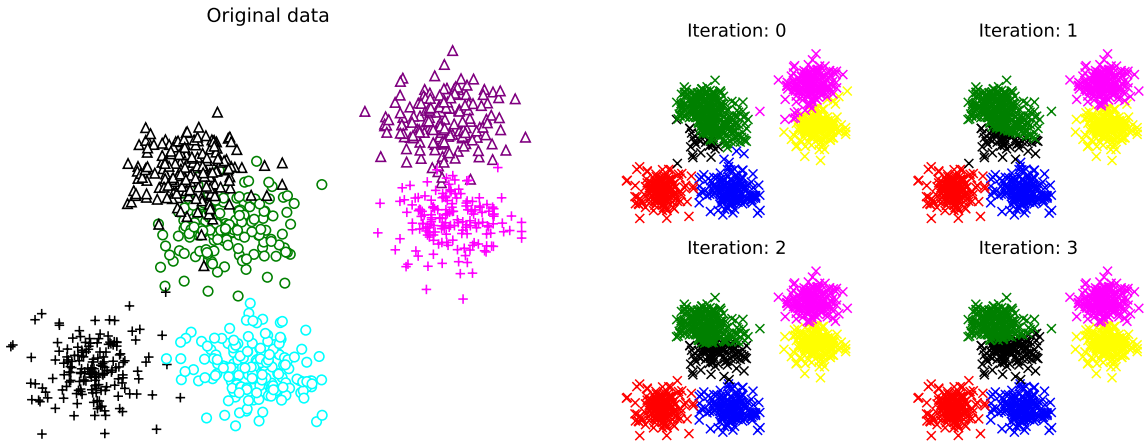

Figure 1: Four-step visualization of K-means.

In this figure, the original classes of the data objects are given in a separate figure window (left part of Figure 1) as well as the cluster-assigned data objects within the first four iterations are also given(right part of Figure 1). To accomplish this objective, follow the procedure below[2]:

1. Load a ready-to-use dataset (D), or generate on your own. Be aware that there are more than five classes where data objects belong to.
2. Show data objects with its original class distribution in a plot window as shown in Figure 1. Then, save it as 'OriginalData.pdf' and **close**[3] figure window.

[2]The exact procedure is not given in this final assignment because you are expected to come up with your own strategy.

[3]**never forget to close your window**; otherwise your work will be penalized!

3. As for the demonstration of update of the cluster centroids as well as the assign of data objects to these clusters. This demonstration should include first four iterations by generating 2×2-axis figure as shown in Figure 1. Note that, the number of cluster to be set for K-means algorithm must be equal to the number of distinct classes in D.

4. Save this figure as 'KmeansDemonstration.pdf' and **close**[3] figure window.

2.2 Visualization of convergence of cluster centroids

Here you are to demonstrate how centroids of the clusters converge to the optimal (or suboptimal) points in the search space from the very first iteration to the last iteration. This task enables you to observe how early K-means algorithm can find its solution[4]. A demonstration for this is given in Figure 2.

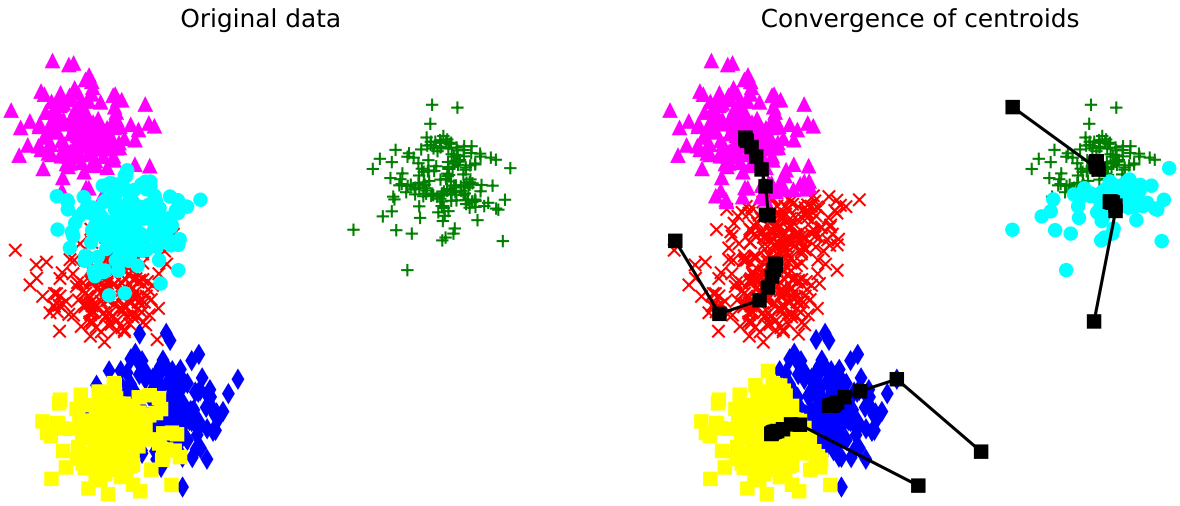

Figure 2: Convergence of clusters' centroids.

As you can see from this figure, randomly initialized centroids converge step-by-step to their optimal points in two dimensional space through the running algorithm (i.e., K-means clustering). Moreover, it can be deduced from the outcome, K-means could find a suboptimal solution. To accomplish this task, follow the procedure below[2]:

1. Load a ready-to-use dataset (D), or generate on your own. Be aware that there are more than five classes where data objects belong to.

2. Apply K-means clustering algorithm with a cluster size equal to the number of classes in D.

3. Generate 1×2-axis figure and show data objects with its original class distribution in the first plot window as shown at left side of Figure 2. As for the second plot window, show convergence of clusters that K-means obtained in each iteration as shown at right side of Figure 2. Be aware that cluster centroids at each iteration are shown by square markers whereas convergence between every controid is indicated by a solid black line.

[4] it is certainly true for few dimensional search space and may take longer time for higher dimensions

4. After you show the convergence save it as 'ConvergenceCentroid.pdf' and **close**[3] figure window.

2.3 Clustering as a means of sampling: Cluster sampling

In this task, you are going to make use of clustering algorithm as a means of sampling which is an important procedure on preprocessing the data.

Cluster sampling:

This approach relies on a procedure that involves grouping a larger population into a number of categories and then selecting only some among them. Therefore, the outcome with strategy is to retrieve only a part of all individual data objects preserving the representation of all data objects. There are different types of this strategy: *single-stage*, *double-stage*, and *multi-stage* sampling:

- **Single-stage sampling:** This is the simplest way for cluster sampling. In this adoption, the whole data set is divided into a number of clusters. The quality of the clusters and how well they represent the larger population determines the validity of your results with this sampling type. That's why, setting a large number for clusters seems to be ideal.

  After categorization of the data object, some of the clusters are chosen for the inclusion of sampled data objects. In standard, a random selection is used; however, because it is expected from this sampling strategy that *i*) each cluster's population should be as diverse as possible and *ii*) each cluster should have a similar distribution of characteristics as the distribution of the population as a whole, it is better to first eliminate clusters having higher density so that each cluster itself is a mini-representation of the larger population. Finally, all the data objects belonging to the selected clusters are taken into consideration.

- **Double-stage sampling:** This strategy applies all the steps in single-stage sampling except for the final step. Here, rather than collecting data objects from every selected clusters, random selection of individuals is applied within the cluster to use as sampled data objects. This strategy allows a more reduced data objects and hence leads a shorter learning time but may lead to a further degradation on the performance. In the case this elimination procedure continues for two or more times it is then called as **multi-stage sampling** strategy.

2.3.1 Single- and double-stage sampling

In this task, you are expected to implement single- and double-stage sampling. The goal of this task is twofold: *i*) visualization of single- and double-stage sampling, and *ii*) analyzing the effect of these strategies on classification task. Visualization of cluster sampling is exemplified in Figure 3 and the results of analysis with respect to the classification accuracy and training time are given thereafter. From the analysis, it can be seen that the classification performance on testing data slightly degrades when cluster sampling is applied but yields a shorter training time which is quite normal due to the reduced data objects. Therefore, data scientists should

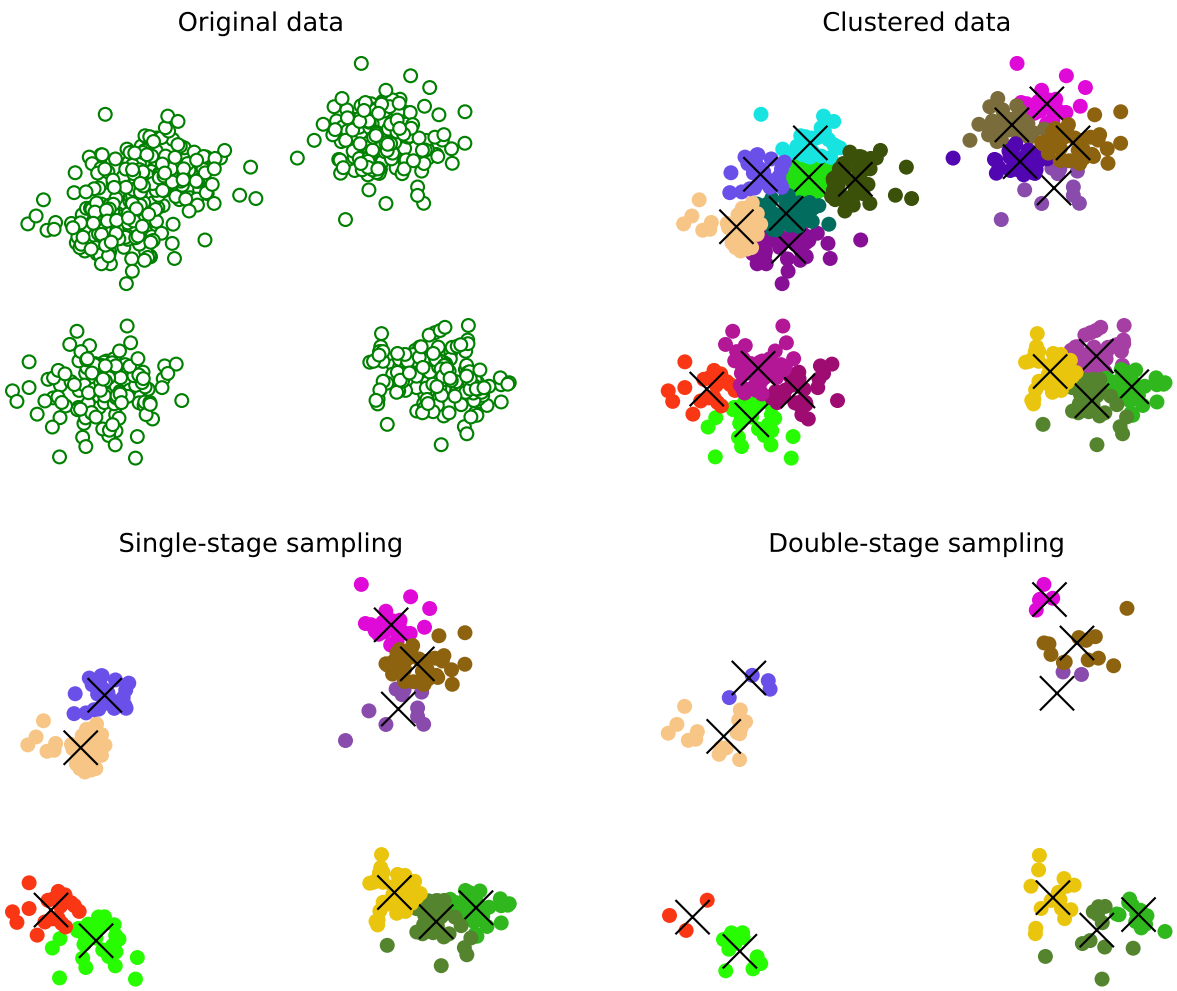

Figure 3: Cluster sampling visualization.

consider a trade-off between performance and learning time depending on the problem they handle with.

```
(Original Data) Mean Testing Accuracy: 0.987 Training Time: 638.899 ms
(Single-stage Clustering) Mean Testing Accuracy: 0.923 Training Time: 300.056 ms
(Double-stage Clustering) Mean Testing Accuracy: 0.903 Training Time: 167.898 ms
```

As for the visualization, it can be seen that data objects that are given at the upper-left part of the figure are initially categorized into a number of clusters, which is shown at the upper-right part of the figure. Single-stage and double-stage sampling outcomes are given in order at the lower part of the figure. As seen, the denser the data clusters is, the more likely they are to be removed for single-stage sampling so that remaining data objects are a mini representation of whole data. As for double-stage clustering some of the individual objects from the chosen clusters are randomly chosen. To accomplish that please follow the procedure with a high care[2]:

1. Load a ready-to-use dataset (D), or generate on your own. Be aware that there are

more than five classes where data objects belong to.

2. Split D such that randomly selected 70% tuples are used for training while 30% tuples are used for testing.
3. Generate 2×2-axis figure and show all the **training** data objects in the upper-left window.
4. Apply K-means clustering algorithm on **training data** with a large number of clusters[5] (it is 20 in Figure 3) and then show the cluster centroids as well as cluster-assigned training data objects in upper-right window.
5. Determine number of clusters (R) to be included for sampling[5] (it is 10 in Figure 3).
6. Select clusters in which data objects are to be included for sampling. Note that, you should remove dense clusters on priority. To do that, you can follow the steps below:
   a) Include all cluster centroids C.
   b) Calculate the density of every centroid ($c \mid c \in C$) by equation below.

$$density_c = \frac{1}{\sum_{n \in N(c)} d(c,n)/||N(c)||} \tag{1}$$

   where $N(c)$ is a set of neighbor cluster centroids of cluster c. The number of neighbor clusters should be determined first[5]. It is set as five in Figure 3.

   c) Remove the cluster having highest density from C then go to step b until R clusters remain in C; otherwise go to following step.
7. Once you reach this step, you must have obtained single-stage sampled data objects (say $train_{ss}$). Visualize $train_{ss}$ at the lower-left part of the figure.
8. To obtain double-stage sampled data objects (say $train_{ds}$), randomly choose data objects from $train_{ss}$. Visualize $train_{ds}$ at the lower-right part of the figure.
9. Once you complete plotting task, save it as 'ClusterSampling.pdf' and **close**[3] figure window.
10. Generate an instance of Multi Layer Perceptron (MLP) classifier with a parameter setting[5].
11. Train MLP with same settings on *original training data*, $train_{ss}$, and $train_{ds}$. Also record training time separately taken by every training phase.
12. Evaluate the performance of models learned on three different data sets through test data that is generated at step 2.
13. Print your findings in a format that exactly matches with given above.

[5] feel free to give any number.

3 Outlier Detection Task

The final objective of this assignment is given for you to gain knowledge on how outliers from a given data set can be detected. As stated earlier, different outlier detection approaches exist in data science. In this task, you are expected to implement your own clustering-based outlier detection algorithm utilizing K-means algorithm.

In this approach, distance of a given point to the closest centroid is an indicator for the detection (it is called as prototype-based). An exemplar to this approach is given in Figure 4. Here, all the **normal** data objects are first grouped into a number of clusters. Then outlier points are separately tested considering the distance between them and the centroid of the closest cluster.

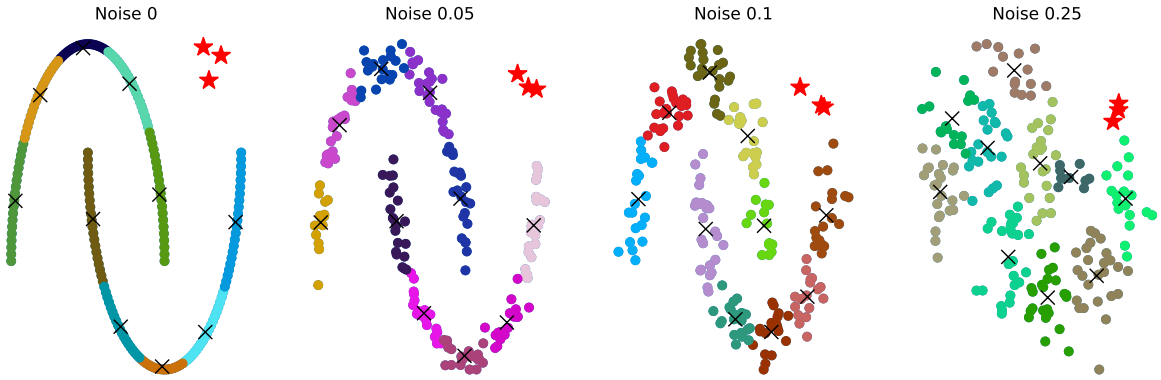

Figure 4: Visualization of outlier detection through clustering.

To accomplish this task, follow the procedure below[2]:

1. For each noise level (n) in $[0, 0.05, 0.1, 0.25]$; repeat the steps below.
    a) Load moon dataset (D) with a noise level of n.
    b) Apply K-means clustering algorithm with a cluster size arbitrarily given (it is 10 in Figure 4).
    c) Calculate the *threshold* that is used for comparison in outlier detection. This value should be equal to the maximum of the distances between all the objects and the cluster centroids that data objects belong to.
    d) Randomly select five data points from D and testify whether they are outlier or not.
    e) Generate three outlier points such that they are apparently far away from the normal data (red star markers in Figure 4); then testify whether they are outlier or not.
    f) The comparison is simple. If the distance between the tested object and closest centroid is larger than the threshold then it is interpreted as outlier and, in such case, it should be displayed with a message `It is outlier!`; otherwise it should be `It is normal.`.

g) You should give an output with a format that should exactly match with the one given below. In the given example, the first five lines are the testing results of normal data objects; while remaining three lines are of outlier data points.

2. Generate 1×4-axis figure and show normal data objects with assigned clusters as shown in the Figure 2. Also show the outlier points with a markers same as those in the figure.

3. Save your plot window as 'OutlierDetection.pdf' and **close**[3] figure window.

```
Setting Noise: 0
It is normal.
It is normal.
It is normal.
It is normal.
It is normal.
It is outlier!
It is outlier!
It is outlier!
Setting Noise: 0.05
It is normal.
It is normal.
It is normal.
It is normal.
It is normal.
It is outlier!
It is outlier!
It is outlier!
Setting Noise: 0.1
It is normal.
It is normal.
It is normal.
It is normal.
It is normal.
It is outlier!
It is outlier!
It is outlier!
Setting Noise: 0.25
It is normal.
It is normal.
It is normal.
It is normal.
It is normal.
It is normal.
It is normal.
It is outlier!
```

Notes

- Your source code should be designed as **easy-to-follow**. **Place comment** in it as much as possible. **Separate each task** through apparent patterns.
- Use LaTeX to prepare your reports. Include the observation tables here to your report. Once again, filled and signed declaration form should be first page of your report. **Reports must not exceed 5 pages in total**.

- **Do not miss** the deadline.
- **Save your work** until the end of this semester.
- The assignment must be **original**, **individual work**. **Duplicate or very similar assignments are both going to be considered as cheating.**
- You can ask your questions via **Piazza** (https://piazza.com/mu.edu.tr/fall2020/ceng3521) and you are supposed to be aware of everything discussed in Piazza.
- You will submit your work on CENG3521 course page at https://dys.mu.edu.tr with the file hierarchy as below[6]:

→ <student id>.zip
→ Assignment4.py
→ Report4.pdf

[6]do not place any file into a directory. Just compress all the files together.