HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
INSTITUTE OF INFORMATION TECHNOLOGY AND COMMUNICATION

MINI-PROJECT REPORT

TOPIC: Visualization of clustering algorithms

(K-Means, K-Nearest Neighbors, Mean Shift Clustering)

Course name: Object Oriented Language & Theory (Lab)

Lecturer: Mrs. Bui Thi Mai Anh

Group 7 – Members Info:

Name	***Student ID***
Trịnh Thùy Giang	20194752
Nguyễn Phương Thảo	20194851
Phạm Thái Duy	20194749

Ha Noi, 4/2021

Table of Contents

PROJECT OVERVIEW

Project topic: Visualization of 3 clustering algorithms (K-Means, K-Nearest Neighbors, Mean Shift Clustering)

Overview: This is a project by Group 7 – class code 122155 – course Object Oriented Language & Theory (Lab). The table below shows the information of group members, including the role of each member in the project.

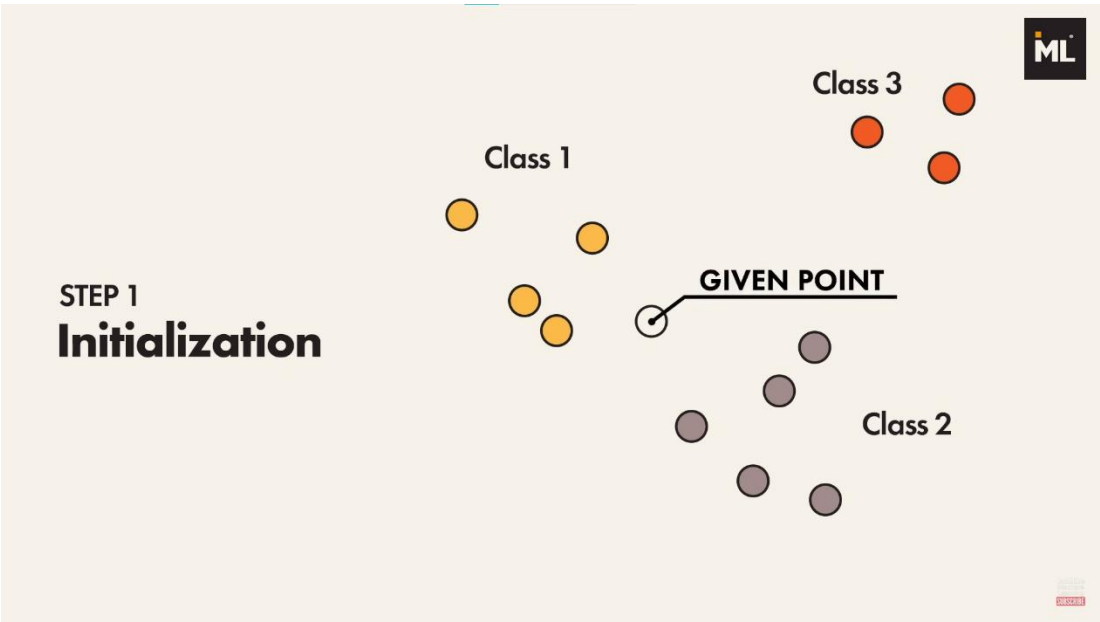
Name	***Student ID***	***Role***
Trịnh Thùy Giang	20194752	- Leader - Design the program - Implement the program menu and meanshift visualization
Nguyễn Phương Thảo	20194851	- Member - Make class diagram - Implement K-nearest neighbors visualization - Write report
Phạm Thái Duy	20194749	- Member - Implement K-means visualization - Presenter

In this project, we'll be showing our class design (class diagram) for the application and also a program user manual. We have also summerized our understanding of the three clustering algorithms, about how they work, problems we target to solve with these algorithms, pros and cons of each algorithm.

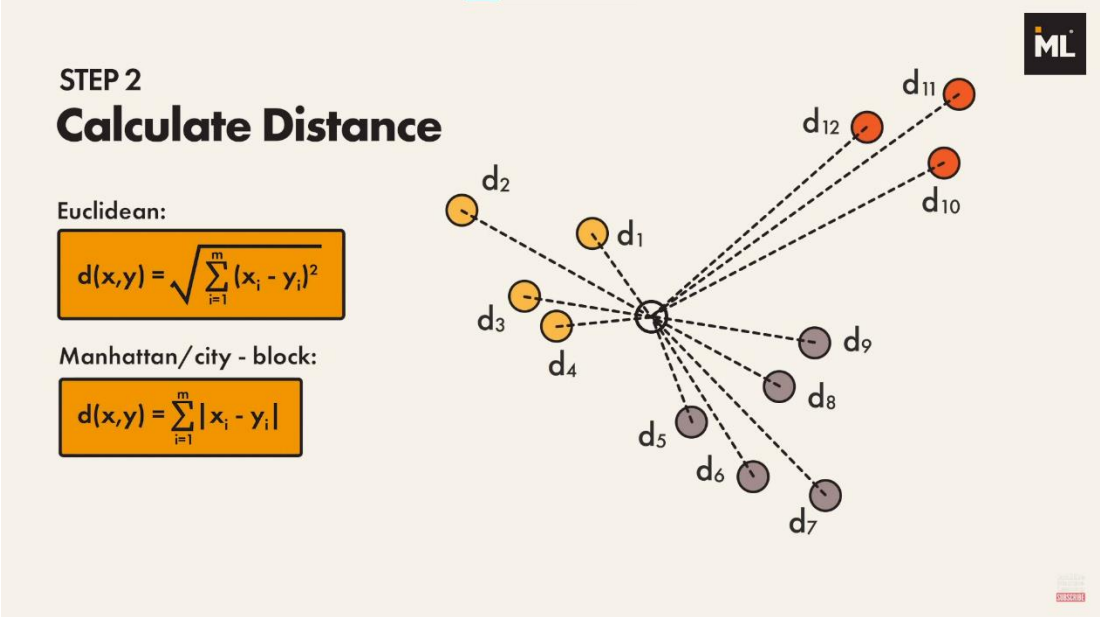
CLUSTERING ALGORITHMS

K-Nearest Neighbors (KNN)

KNN is a super simple supervised machine learning algorithm that can be solved for both classification and regression problem. In this project, we will focus mainly on classification problem. To have a better understanding of this algorithm, let's take a look at the following simple two-dimensional example.

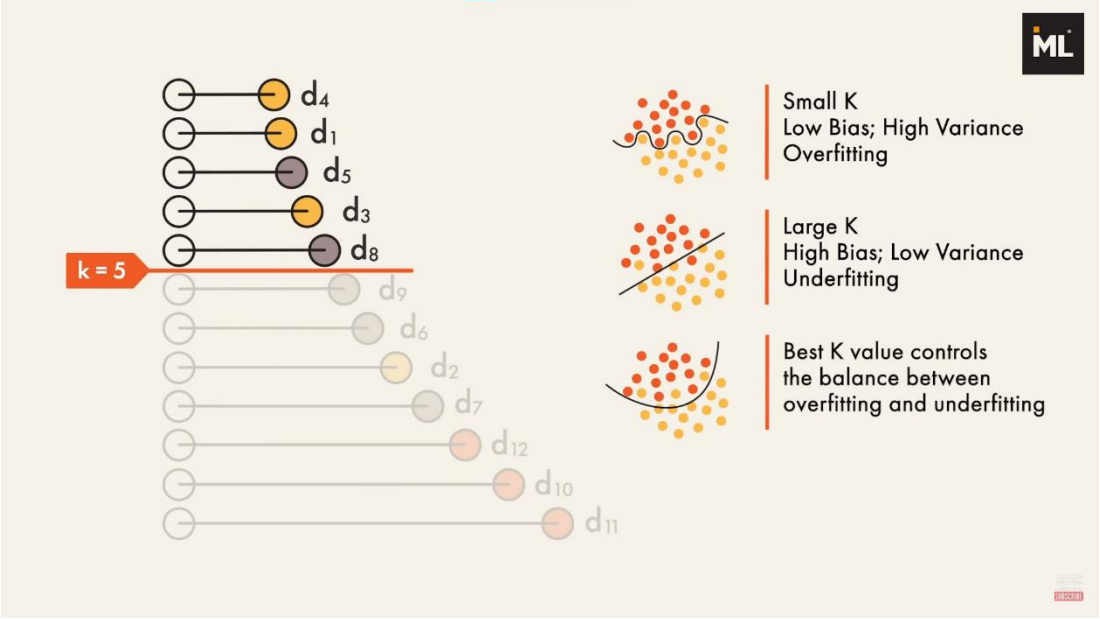


Let's say we want to classify the given point into one of the three groups. In order to find the K nearest neighbors of the given point, we need to calculate the distance between the given point to the other points. There are many distance functions but the Euclidean is the most commonly used one.



Then we need to sort the nearest neighbors of the given point by the distances in increasing order. For the classification problem, the point is classified by vote of its neighbors, then the point is assigned to the class most common among its K nearest neighbors. (in the picture below, K = 7)

K value here controls the balance between overfitting and underfitting, the best value can be found with cross validation and learning curve. A small K value usually leads to low bias but high variance, and a large K usually leads to high bias but low variance, it is important to find a balance between them.



Pros:

- *Extremely easy to implement*
- *It is lazy learning algorithm -> requires no training prior to making real time predictions. This makes the KNN algorithm much faster than other algorithms that require training e.g SVM, linear regression, etc.*
- *Since the algorithm requires no training before making predictions, new data can be added seamlessly.*

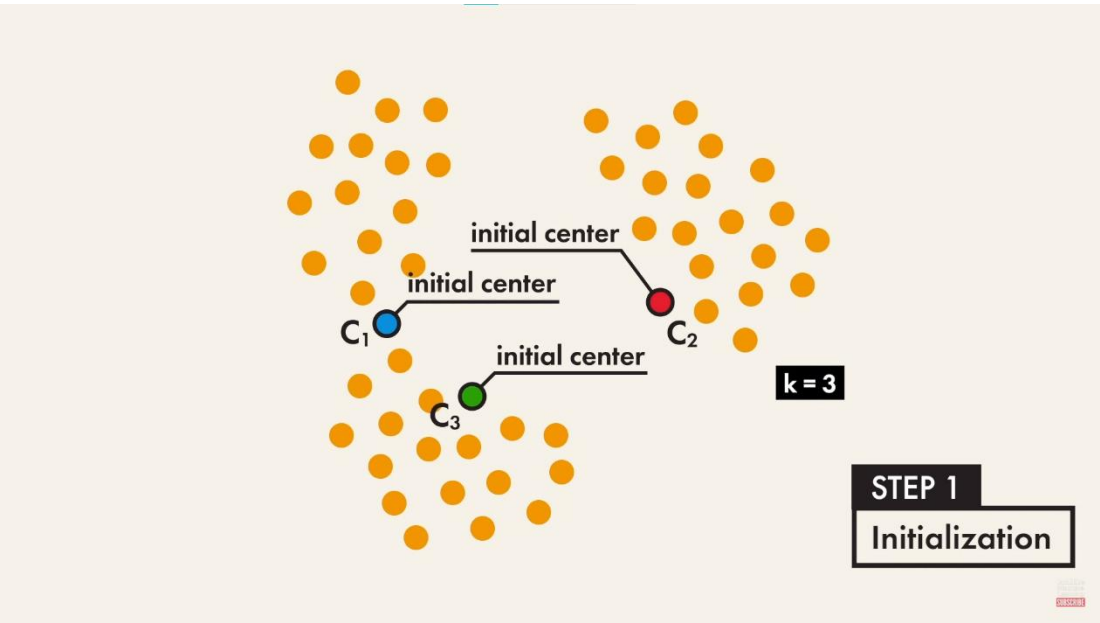
- *There are only two parameters required to implement KNN i.e. the value of K and the distance function (e.g. Euclidean or Manhattan etc.)*

Cons:

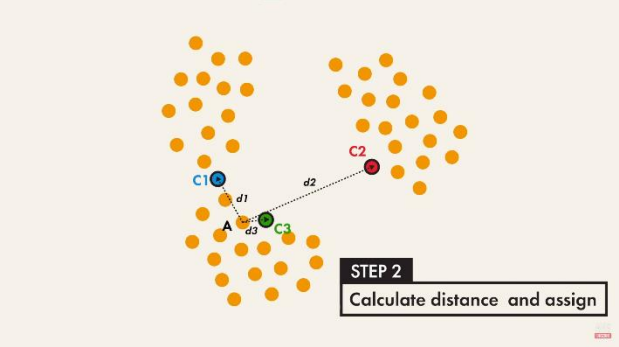
- *Doesn't work well with high dimensional data because with large number of dimensions, it becomes difficult for the algorithm to calculate distance in each dimension.*
- *A high prediction cost for large datasets. This is because in large datasets the cost of calculating distance between new point and each existing point becomes higher.*
- *Doesn't work well with categorical features since it is difficult to find the distance between dimensions with categorical features.*

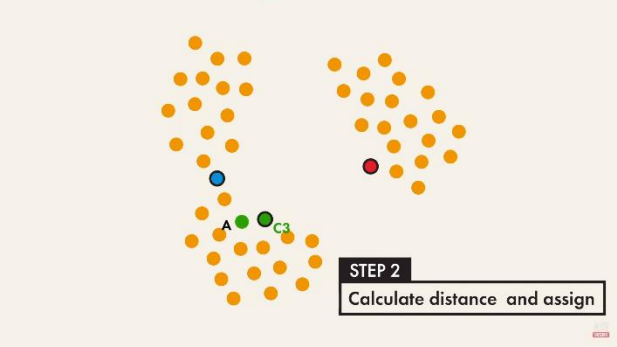
K-Means

K-means is very simple, but also very effective unsupervised clustering algorithm. It tries to partition the datasets into K pre-defined clusters where each data points belong to only one group. Let's go through a simple two-dimensional example to have a better understanding of this idea.

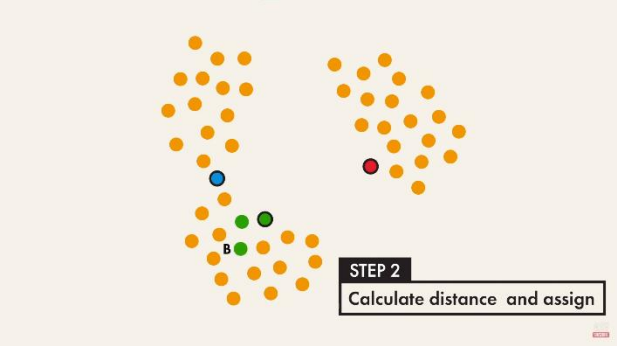


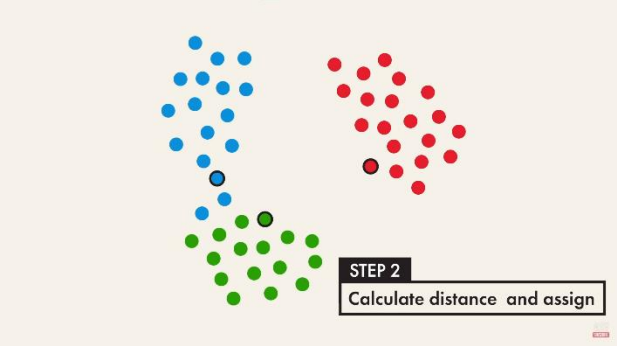
There are 3 major steps. In step 1, we randomly choose K data points from the data set as initial center. In this case, K = 3



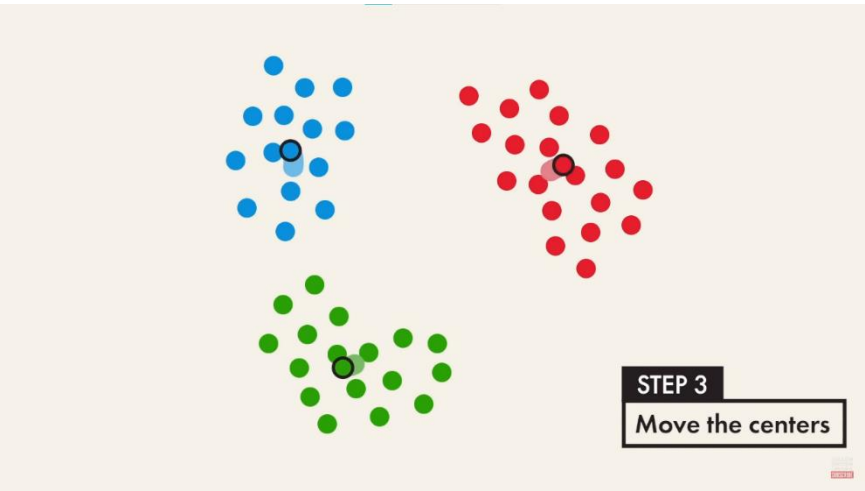


Once we have these centers, in step 2, we can calculate the distance between the data points to these centers. Then we can assign each data points to the closest centers. In this example, for point A we compute its distance to C1, C2 and C3, respectively. After comparing the length of d1, d2 and d3, we can easily notice that d3 is the smallest. Therefore, we assign A to C3. We then move to point B and follows the same procedure. After that we keep repeating this process until all the data points are assigned.





Then we calculate the center's new position by taking the average of all the points within the cluster. Keep repeating this process until all the centers stop moving.



Pros:

- *Easy to understand and implement.*
- *If we have large number of variables then, K-means would be faster than Hierarchical clustering.*
- *On re-computation of centroids, an instance can change the cluster.*
- *Tighter clusters are formed with K-means as compared to Hierarchical clustering.*

Cons:

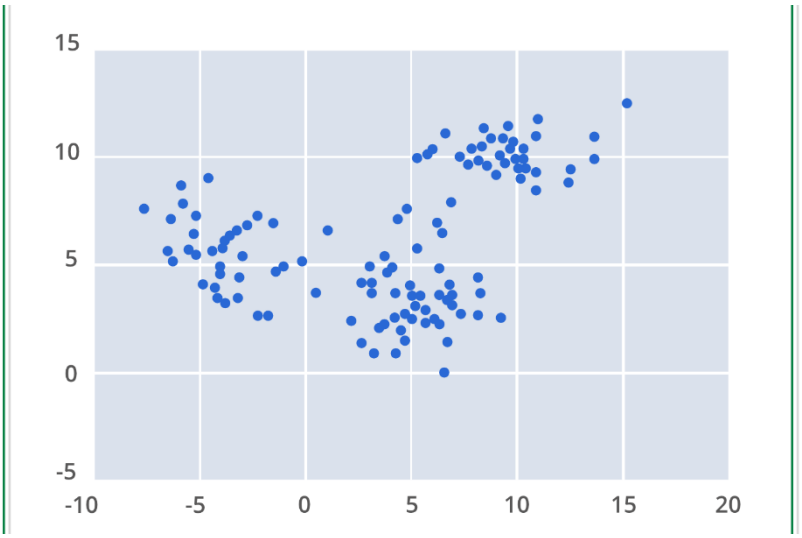
- *It is a bit difficult to predict the number of clusters i.e. the value of k.*
- *Output is strongly impacted by initial inputs like number of clusters (value of k).*
- *Order of data will have strong impact on the final output.*
- *It is very sensitive to rescaling. If we will rescale our data by means of normalization or standardization, then the output will completely change.final output.*
- *It is not good in doing clustering job if the clusters have a complicated geometric shape.*

Mean-Shift Clustering

Mean-shift clustering is another powerful clustering algorithm used in unsupervised learning. Unlike the K-Means cluster algorithm, mean-shift does not require specifying the number of clusters in advance. The number of clusters is determined by the algorithm with respect to the data.

Mean-shift algorithm basically assigns the datapoints to the clusters iteratively by shifting points towards the highest density of datapoints i.e. cluster centroid.

We can understand the working of Mean-Shift clustering algorithm with the help of following steps:



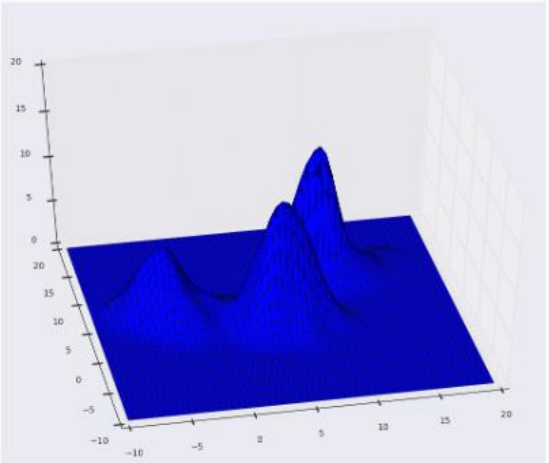
Step 1 − First, start with the data points assigned to a cluster of their own.
Step 2 − Next, this algorithm will compute the centroids.
Step 3 − In this step, location of new centroids will be updated.
Step 4 − Now, the process will be iterated and moved to the higher density region.
Step 5 − At last, it will be stopped once the centroids reach at position from where it cannot move further.

Mean-shift builds upon the concept of kernel density estimation is sort KDE. Imagine that the above data was sampled from a probability distribution. KDE is a method to estimate the underlying distribution also called the probability density function for a set of data.

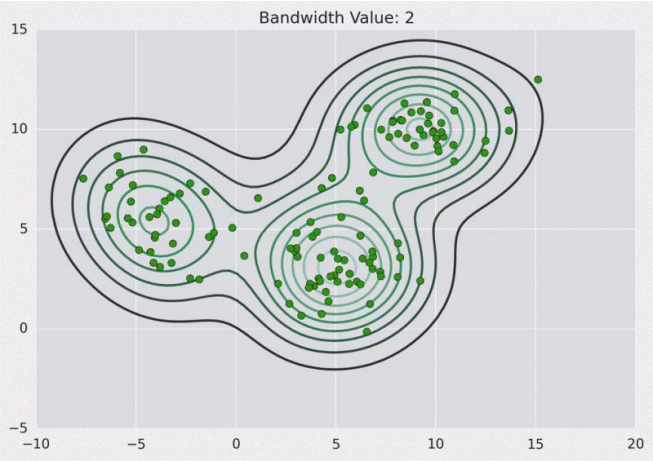
It works by placing a kernel on each point in the data set. A kernel is a fancy mathematical word for a weighting function generally used in convolution. There are many different types of kernels, but the most popular one is the Gaussian kernel. Adding up all of the individual kernels generates a probability surface example density

function. Depending on the kernel bandwidth parameter used, the resultant density function will vary.

Below is the KDE surface for our points above using a Gaussian kernel with a kernel bandwidth of 2.



Surface plot



Contour plot

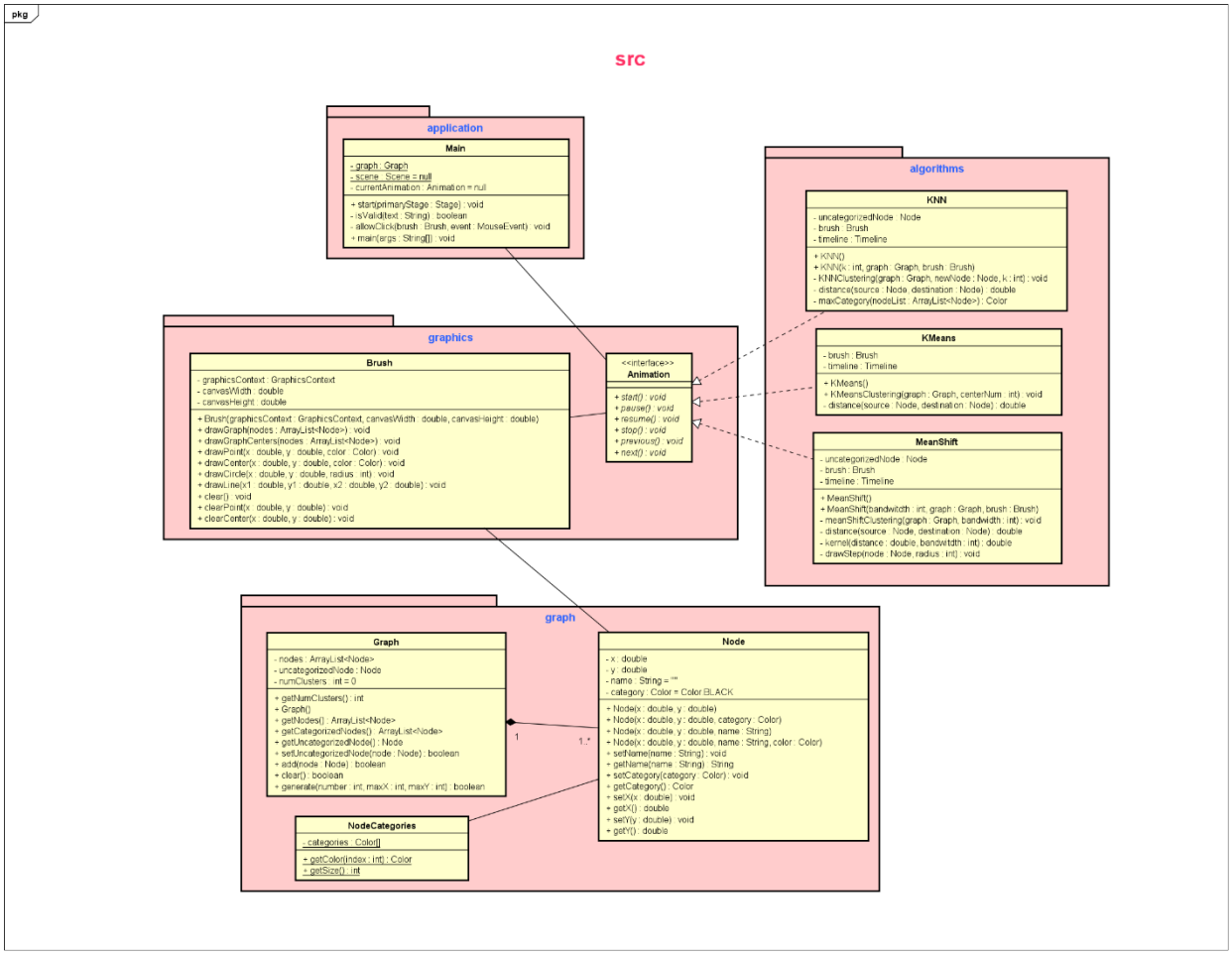
Pros:

- *Finds variable number of modes*
- *Robust to outliers*
- *General, application-independent tool*
- *Model-free, doesn't assume any prior shape like spherical, elliptical, etc. on data clusters*
- *Just a single parameter (window size h) where h has a physical meaning (unlike k-means)*

Cons:

- *Output depends on window size*
- *Window size (bandwidth) selecHon is not trivial*
- *Computationally (relatively) expensive (approx 2s/image)*
- *Doesn't scale well with dimension of feature space.*

CLASS DESIGN



Packages	Classes	Explanation
application	*Main*	*Run the program*
graphics	*Brush*	*Contains tools used to draw shapes, nodes for visualizing*
	Animation (interface)	*An interface contains different types of animations that the program would use while visualizing the algorithms such as play, pause, next, previous,...*
graph	*Graph*	*The program uses a graph as a data set to visualize an algorithm. This class contains attributes of a graph such as nodes, number of clusters, methods like generate a new graph, get number of nodes…*
	Node	*Nodes form a graph, and each node has its own coordinates (x and y) and color to be categorized*
	NodeCategories	*Contains different categories of the node*

algorithms	*KNN*	*K-nearest neighbors algorithm implementation and some specific animations for better visualizing*
	Kmeans	*K-means clustering algorithm implementation and some specific animations for better visualizing*
	MeanShift	*Mean-shift clustering algorithm implementation and some specific animations for better visualizing*

PROGRAM USER MANUAL

Set-up manual

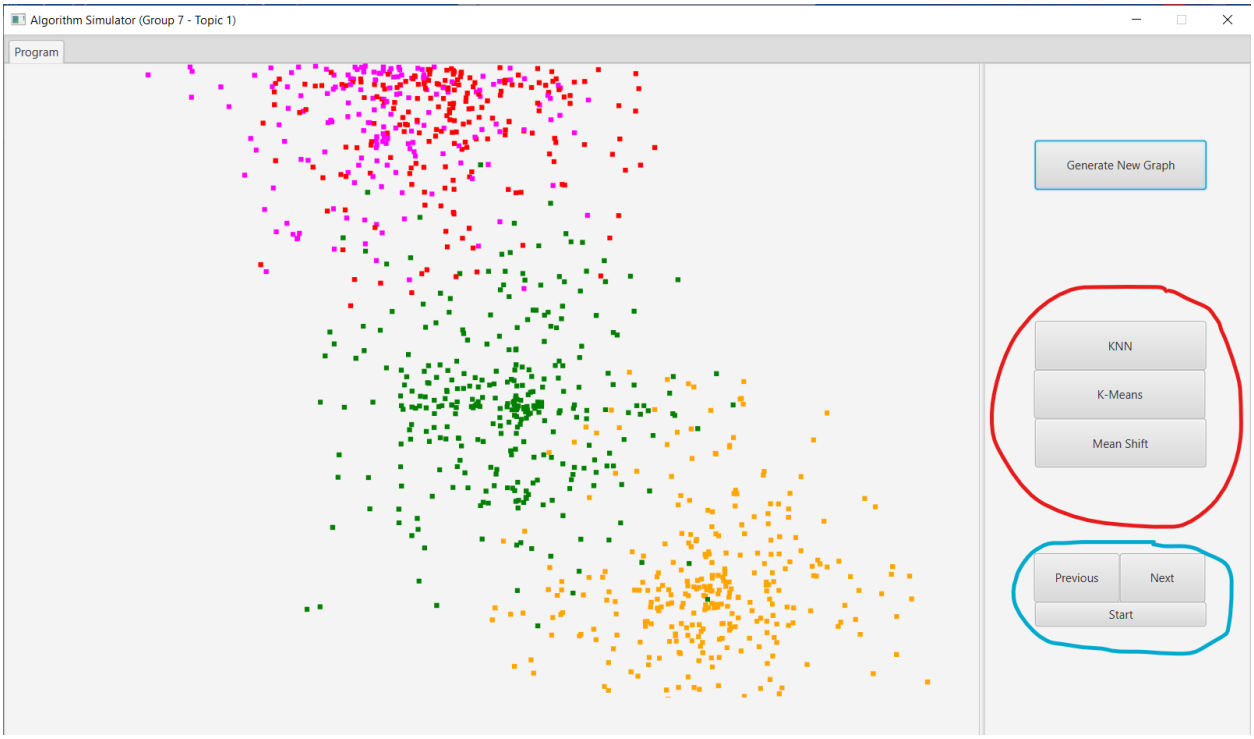
To set-up and run the program from command line, first you need to make sure that your PC/laptop had already installed JDK version 11 or above. Then follow the steps below:

- Step 1: Download the .zip file, then extract the file
- Step 2: Open command prompt window of your device, go to the directory where you store the .jar file (cd “<your directory address>”)
- Step 3: Copy the text in the “run.txt” file to command prompt window to run the program from command line, ***remember to replace the address "D:\Java\javafx-sdk-16\lib" with the address of your javafx-sdk “lib” folder***

Program user manual

- After opening the program, it should look like this:

- First, we need to generate a data set (graph) by click on the button “Generate New Graph” and enter the number of nodes. We can use the same graph for all 3 algorithms.



- To choose an algorithm for the program to visualize, click on 1 of the 3 buttons “KNN” (for K-nearest neighbors visualization), “K-Means” (for K-means clustering visualization), “Mean Shift” (for mean shift clustering visualization). Then enter the required input

- (If you choose button “K-Means” then skip this step) For KNN and Mean Shift, we must draw a node on the graph to be able to run the algorithm. Just click any where in the graph to draw, the node will have the color black (for KNN that is the uncatergorized node, for Mean Shift that is the cluster centroid)

- Click “Start” to see the visualization of the chosen algorithm, this option will run the visualizing non-stop. You can click the “Pause” button to pause the visualization or “Stop” button to end the visualization immediately.

- You can also choose “Previous” or “Next” button to see the visualizing process step-by-step.

- If you want to change to another algorithm visualization while still using the same graph, you must click on the “Stop” button first to stop current algorithm, then you can select other algorithms.

CONCLUSION

This mini-project includes an application to visually demonstrate the three clustering algorithms, which are the K-Nearest Neighbors, K-Means Clustering and the Mean Shift Clustering algorithm. The program is designed and implemented using Java, with graphical user interface based on JavaFX.

- THE END -