University of Canberra
Faculty of Science and Technology

Programming for Data Science 11521G (Online & On-campus)

Assignment 1

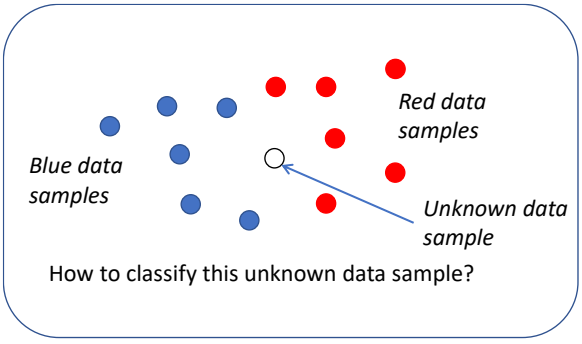
Classifier and Cluster Analysis in Data Science

Due dates: 23:59 Sunday 19/09/2021 (Week 7)
Type: Individual assignment
Marks in this assessment: 100 points (20% of final mark)
Submission: Submit a .zip file containing all Python files (.py) in your project via Canvas site.
Late submission: 5% of the total mark per day (5 marks per day).

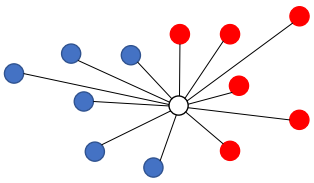
[6 marks] Question 1: Implement a Python program for **Nearest Neighbour Classifier** that can classify an unknown data sample to one of the given classes.

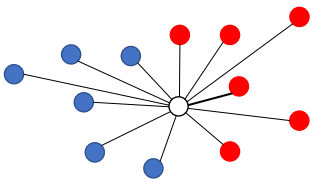
For example, there are 2 classes **Red** and **Blue**, and *x* is an unknown data sample (i.e., we do not know *x* is red or blue). After calculating all distances between *x* and all data samples in the 2 classes, we find a data sample in the Red class that has shortest distance to *x*, so *x* is classified as a red data sample.

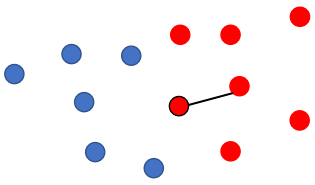
Requirements: Your program reads data samples from 2 text files for 2 classes and unknown data samples from another text file, runs the Nearest Neighbour Classifier algorithm as demonstrated in the screenshots below, and outputs all unknown data samples and their classified label to screen and to another text file. Your program should work with any data dimension D > 1 and any number of unknown data samples > 0. For Python programming, use a **tuple** to store a data sample, a **list** to store all data samples, and **modules** to store functions. The main program includes only function calls and does not include any function implementations. Please do not use other versions of Nearest Neighbour Classifier you can find on websites or research articles, and do not import any external packages (except **tkinter**) to this project.



How to classify this unknown data sample?


Step 1. Calculate all distances between this unknown data sample to all other data samples


Step 2. Find the minimum distance

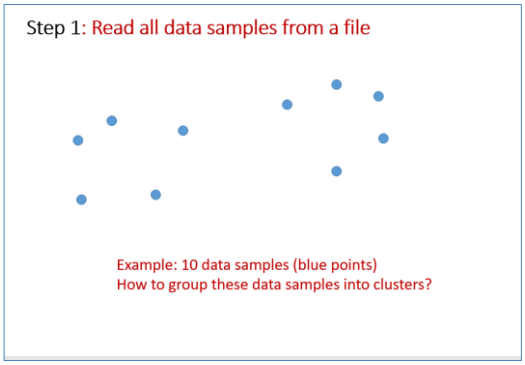

Step 3. Output result: the unknown data sample is red

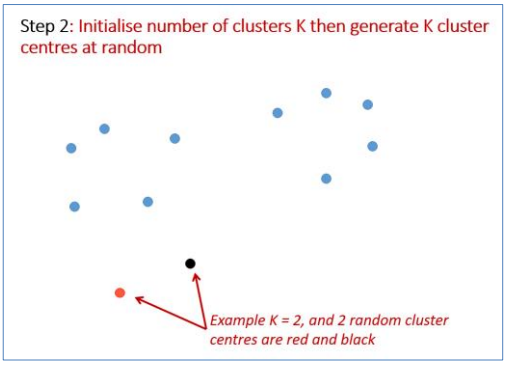
[14 marks] Question 2: Implement a Python program for **K-Means Clustering** that can group data samples to clusters.

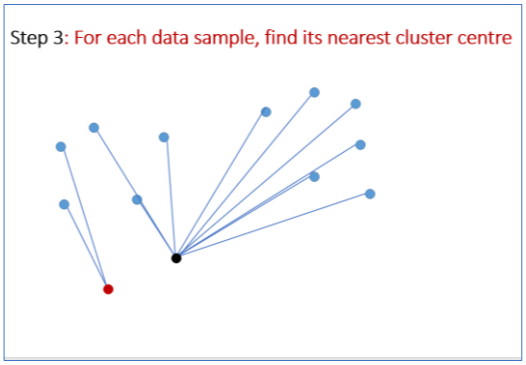
For example, you are given a set of data samples to group them into 2 clusters. The K-means clustering algorithm generates 2 cluster centres at random, groups data samples that are nearest to the first cluster centre to form a cluster then do the same with the second one to form another cluster. The algorithm will generate new cluster centres by averaging data samples in the same cluster. If the difference between the 2 old cluster centres and the 2 new cluster centres are not significant, the algorithm will stop, otherwise it removes the old cluster centres and re-groups data samples for the new cluster centres as seen above to form new clusters. The process repeats until the difference between the old and new cluster centres is not significant.

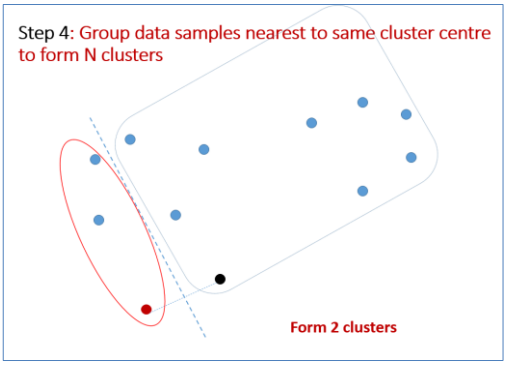
Requirements: Your program reads data samples from a text file, runs K-means Clustering algorithm as demonstrated in the screenshots below, and outputs all data samples with cluster centres to screen as below. Your program should work with any data dimension D > 1 and any number of clusters K > 1. For Python programming, use **tkinter** to display data samples and cluster centres on a canvas, a **tuple** to store a data sample or a cluster centre, a **list** to store all data samples or all cluster centres, and **modules** to store functions. The main program includes only function calls and does not include any function implementations. Please do not use other versions of K-Means Clustering that you can find on websites or research articles to implement this project. Please do not import any external packages (except **tkinter**) to this project.

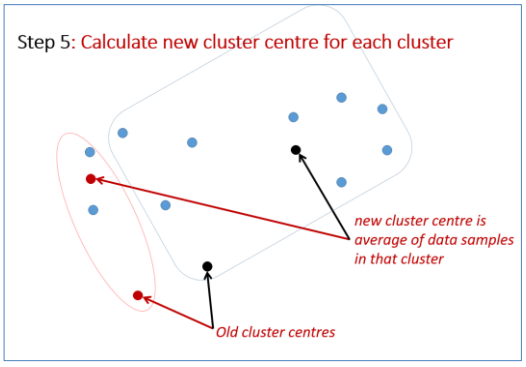
The screenshots below explain how K-means Clustering algorithm works.

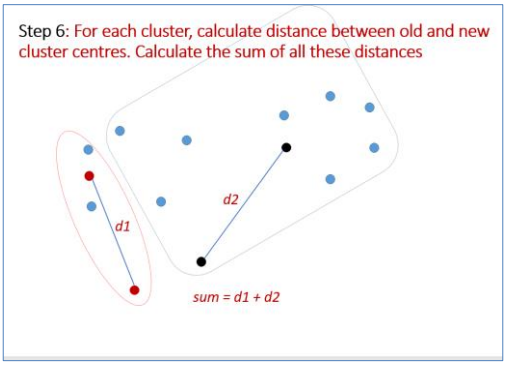


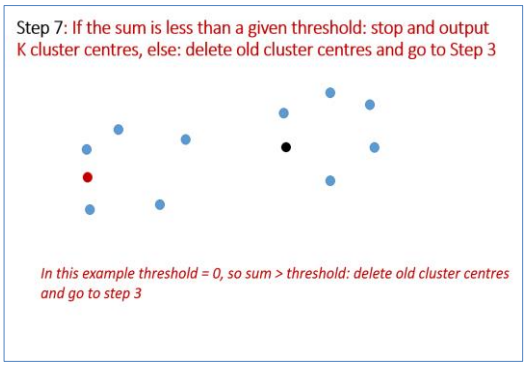


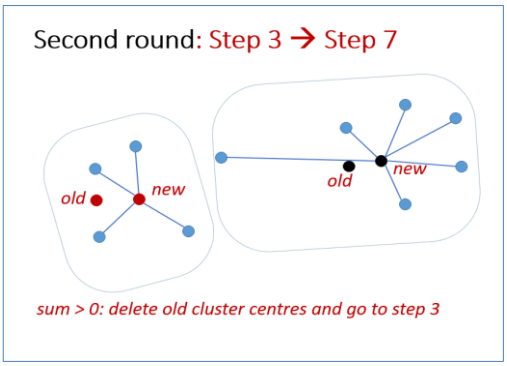


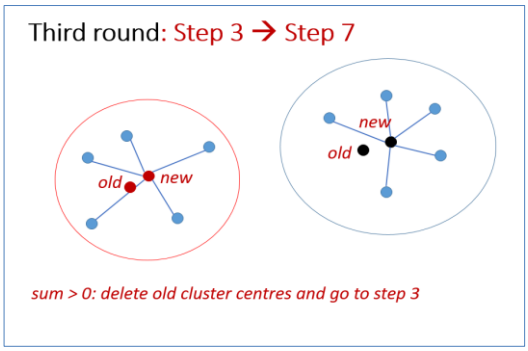


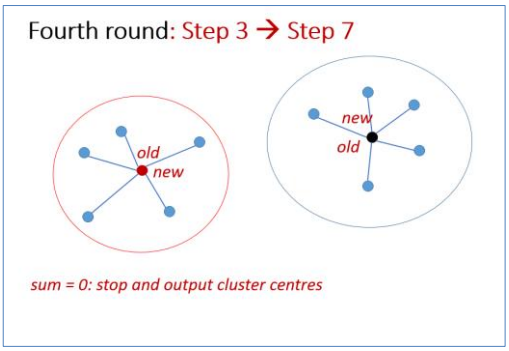




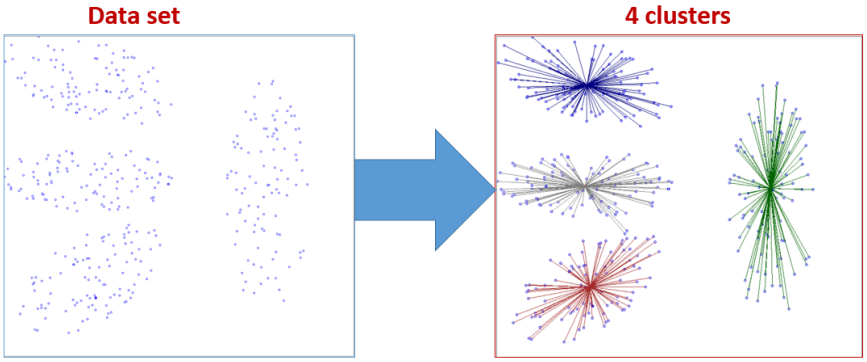








Below is an example of data samples drawn on screen before and after applying K-means clustering.



More details of the above algorithms and demos will be given in lectures and tutorials from Week 2 to Week 7.

-- END –

Assignment 1 Marking Guideline

Question 1 [30 marks]

- [2.5 marks] Read 3 files for red, green, and unknown data sets
- For each unknown sample in the unknown data set
    - [2.5 marks] Calculate distances from the unknown sample to all red data samples
    - [2.5 marks] Find min_1 (minimum distance of the above distances to red samples)
    - [2.5 marks] Calculate distances from the unknown sample to all blue data samples
    - [2.5 marks] Find min_2 (minimum distance of the above distances to blue samples)
    - [2.5 marks] Compare min_1 and min_2 and assign class label to the unknown sample
- [2.5 marks] Output all unknown samples and their class label to screen
- [2.5 marks] Output all unknown samples and their class label to file
- [2.5 marks] Data sample is tuple, red, blue and unknown data samples are stored in 3 lists
- [2.5 marks] All functions are in a module file, no function is in main program
- [2.5 marks] Exception handling
- [2.5 marks] Overall (Output on canvas and Python code writing)
- [– 10 marks] The program cannot work with any number of dimensions
- [– 10 marks] External packages imported (except tkinter)
- [– 10 marks] Algorithm is quite different from the given algorithm
- [– 5 marks] Lack of comments that explain your code

Question 2 [60 marks]

- [5.25 marks] Read data file, get number of dimensions D and number of data samples N
- [5.25 marks] Input number of clusters K, create K clusters same dimension D at random, and set threshold to a small value
- Repeat the following:
    - [5.25 marks] For each data sample, find its nearest cluster centre
    - [5.25 marks] Group data samples having the same nearest centre to a cluster
    - [5.25 marks] For each cluster, calculate new cluster centre (average of all samples)
    - [5.25 marks] Calculate sum of distances between old and new cluster centres
    - [5.25 marks] If the sum is less than the threshold: display K cluster centres and data samples on canvas then break, else: set cluster centres to new cluster centres
- [5.25 marks] Data sample is tuple, all data samples are stored in a list
- [5.25 marks] All functions are in a module file, no function is in main program
- [5.25 marks] Exception handling
- [7.5 marks] Overall (Output on canvas and Python code writing)
- [– 20 marks] The program cannot work with any number of dimensions
- [– 20 marks] External packages imported (except tkinter)
- [– 20 marks] Algorithm is quite different from the given algorithm
- [– 10 marks] Lack of comments that explain your code