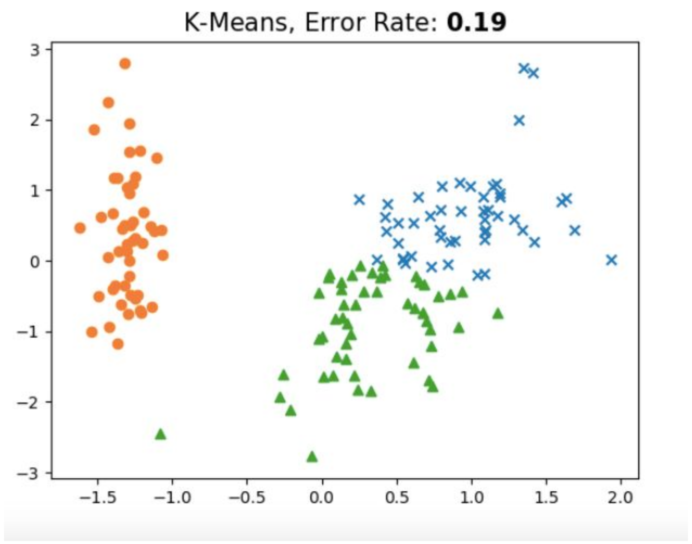
Lab 2

It is an individual programming assignment. This lab assignment is graded based on 100 points and is an individual effort (e.g, no teamwork is allowed).

Part 1 (10x6 = 60 pts) KNN & K-mean

For the datasets provided, complete all the steps requested: **NOTE: You should use both your own NumPy implementation of the algorithms and sklearn.**
Step 1: Load the data points and ground truth values (Some ground truth values may be in a different file)
Step 2 (1 pts): Plot the points according to the ground truth value classes. For example if there are 3 different classes, expected output:



Step 3: Split data into train and test.
Step 4 (2 pts): Use KNN **(own NumPy code & sklearn library)** to clusters the data.
Step 5 (1 pts): Plot your result in different colors similar to Step 2.
Step 6 (1 pts): Print the confusion matrix to check model performance.
Step 7 (4 pts): Repeat Steps 3 - 6 and use K-means **(own NumPy code & sklearn library)** to perform the same. Additionally, print the centroids.
Step 8 (1pts): Compare the results between KNN and K-means

Datasets:

1. **Dataset_1/S1.txt:** N=5000 vectors and k=15 Gaussian cluster
2. **Dataset_1/S2.txt:** N=5000 vectors and k=15 Gaussian cluster
3. **Dataset_1/S3.txt**: N=5000 vectors and k=15 Gaussian cluster
4. **Dataset_1/S4.txt:** N=5000 vectors and k=15 Gaussian cluster

NOTE: Ground Truth and original centroid for S1-4 are present in the **dataset_1/s-originals** folder.

5. **Dataset_2/dim32.txt**: 32 dimensions;
6. **Dataset_3/spiral.txt:** N=312, k=3, D=2; Ground truth is the third column present in the same file.

Part 2 (30 pts): Movie Recommender

Context

In this lab, you will be implementing a simple movie recommender system.

Dataset details

You will be using the ml-m1 dataset from the MovieLense website.

You will be using movies.dat and rating.dat for building your recommender.

Steps:

1. Create m x u matrix with movies as row and users as column
2. Normalize the matrix
3. Compute SVD to get U, S, and V.

   Use np.linalg.svd()

4. From your V.T select 50 components
5. Implement a function that takes movieID as input and then implement cosine similarity along with sorting to recommend the top 10 movies.

Project Report (10 pts)

1. For each dataset in part 1, show the two plots (original & your predictions). KNN, K-means comparison with a reason for better performance(if any).
2. For part2, take any 3 random movie IDs. Compute and report the top 10 recommended movies. Describe if the movies are actually similar.

Deliverables

1. Project report
2. **Fully commented** Jupyter notebook