Q1 First note that, the 8 observations with two features $x_1$, $x_2$ clearly forms two clusters, one being at top left, and the other being at bottom right corner.

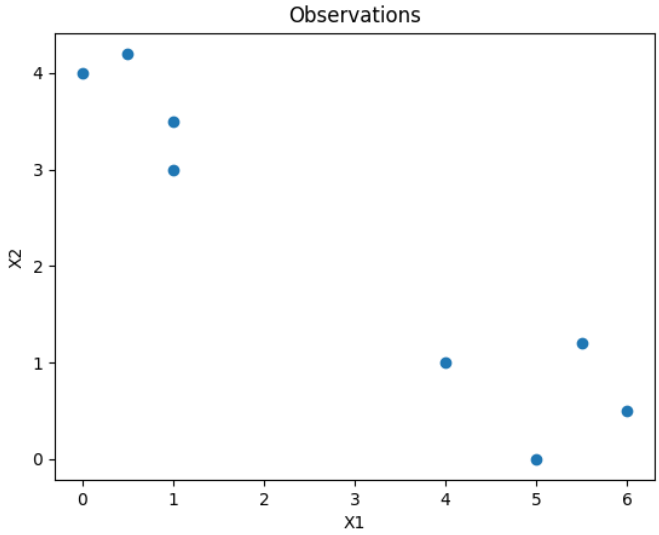


Figure 1: Original data given in the question.

Then K-means algorithm with $k = 2$ is performed manually with the 8 data points as given in the problem. Label '0' and '1' are first randomly assigned, then the location of the centre is calculated for each cluster. Update the centre by reassigning label according to the closest cluster centre for each observation. Repeat until the label is not changing.

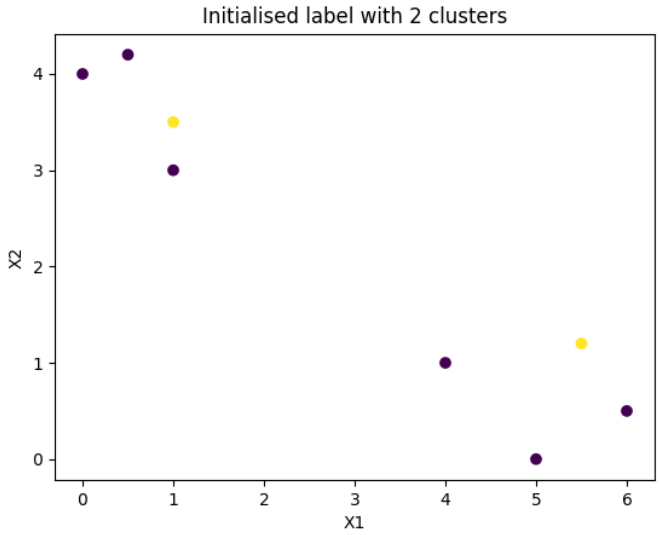


Figure 2: Randomly assigned label, with number of clusters being $k = 2$.

```
centroids = np.zeros((K, p))
flag = True
dist_ = np.zeros((K, n))

while flag:
```

```
# update label
labels = labels_new

# (c) Compute the Centroids
for i in range(K):
     centroids[i] = np.mean(x[:, labels==i], axis=1)

# (d) Reassign Labels According to the Nearest Centroids
for i in range(K):
     dist_[i] = np.sum((x-centroids[i][:, np.newaxis])**2, axis=0)

labels_new = np.argmin(dist_, axis=0)

# continue if label is changing
flag = any((labels - labels_new) != 0)
```

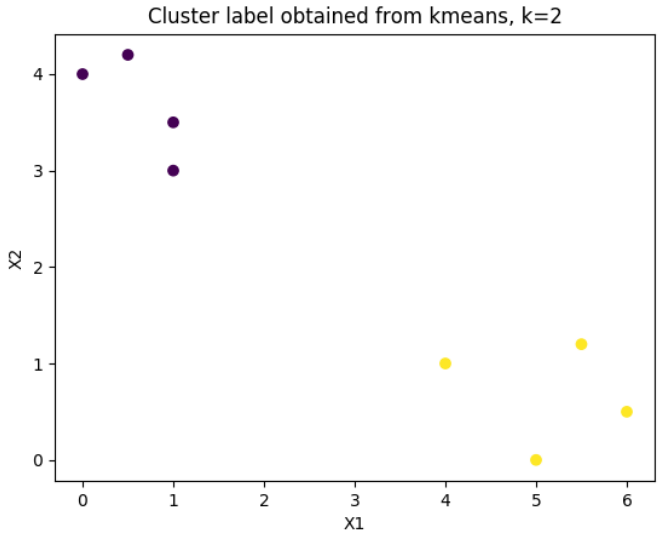


Figure 3: Label given by the K-means algorithm, with $k = 2$.

The final classification in Fig 3 is exactly as expected, two perfectly separated clusters, one at top left corner and the other at bottom right corner.

Q2 (a) For this simulation study, the data to be generated consists of three classes with 25 observations and 50 features respectively. To allow for variability of the distributions of features, each feature is randomly assigned to one of the following: Gaussian mixture with 3 modes, standard Cauchy, unimodal Gaussian, Uniform, Gamma(2,2). For the Gaussian Mixture, the mean and covariance matrix are randomly generated, where the mean is drawn from a Uniform distribution and the Covariance matrix uses function `make_spd_matrix` in `scikit-learn`. Similarly, the mean and standard deviation of Gaussian are also drawn from a Uniform distribution. The visualisation of the generated data with first two features is as in Fig 4.

(b) Then, PCA is performed on the 75 observations with `PCA` function from `scikit-learn`. The first two principal component score vectors are plotted in Fig 5.

(c) k-means clustering is performed with $k = 2$, $k = 3$, $k = 4$ respectively, with `KMeans` function from `scikit-learn`. The classification result is visualised in Fig 6 with first two features.

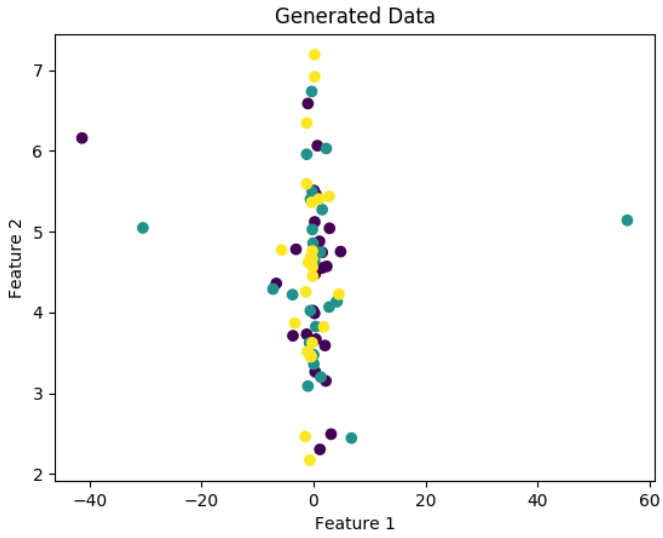


Figure 4: Generated data consisted of three classes with 25 observations and 50 features respectively. First two features are visualised

As we can see from the plot, for each choice of k, the performances of the classification are not good. We further provide the confusion matrix to illustrate the classification result.

| | Predicted, k=2 | | |
|---|---|---|---|
| Actual | 25.00 | 0.00 | 0.00 |
| | 25.00 | 0.00 | 0.00 |
| | 24.00 | 1.00 | 0.00 |

| | Predicted, k=3 | | |
|---|---|---|---|
| | 25.00 | 0.00 | 0.00 |
| | 24.00 | 0.00 | 1.00 |
| | 24.00 | 1.00 | 0.00 |

| Predicted, k=4 | | | |
|---|---|---|---|
| 24.00 | 0.00 | 0.00 | 1.00 |
| 24.00 | 0.00 | 1.00 | 0.00 |
| 23.00 | 1.00 | 0.00 | 1.00 |
| 0.00 | 0.00 | 0.00 | 0.00 |

(f) Instead of performing Kmeans on the original data, we instead perform on the first two principle component of the PCA transformed data. Th result is plotted as in Fig 7, which is identical for the first two principle component.

The confusion matrix below also shows that the classification gives almost the same result as of the generated label, except for one point.

| | Predicted, k=3 | | |
|---|---|---|---|
| Actual | 24.00 | 1.00 | 0.00 |
| | 0.00 | 0.00 | 25.00 |
| | 0.00 | 25.00 | 0.00 |

(g) Now, we perform the Kmeans algorithm on scaled data. After scaling each variable to have standard deviation. Because some of the features are simulated from standard Cauchy distribution, which is likely to give outliers that are not easily detectable by the algorithm if the data is not normalised.

As in Fig 8 and confusion matrix below, the algorithm now perfectly classify the data into the classes as generated. The classification result is indeed improved by homogenising the scale of variances among the features.

| | Predicted, k=3 | | |
|---|---|---|---|
| Actual | 0.00 | 0.00 | 25.00 |
| | 25.00 | 0.00 | 0.00 |
| | 0.00 | 25.00 | 0.00 |

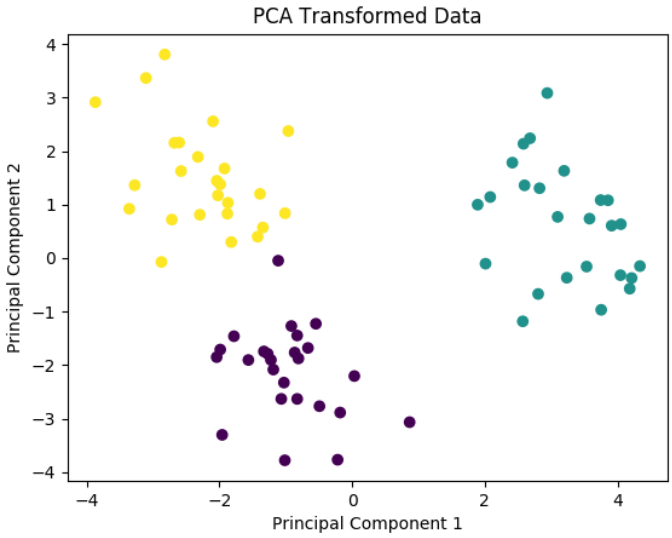


Figure 5: PCA transformed data, visualised with first two principle components.

Q3 We first simulate data as in Question 3, visualised in Fig 9, then a prgramme is written to perform Kmeans clustering by extending the answer to Q1. Especially, when a label has no data being assigned to, the centre location is set to infinity.

As the algorithm runs, the centre of each cluster gets updated and in the case of $k = 3$, the data is being perfectly separated into three clusters as in Fig 10, which roughly resembles the denoised data, i.e. without the inclusion of the Gaussian noise to $x, y, z$.

And the confusion matrix is as below.

| | | Predicted, k=3 | | |
|---|---|---|---|---|
| Actual | 1 | 21 | 5 |
| | 21 | 8 | 7 |
| | 2 | 1 | 18 |

Then we consider the performance of the programme for varying number of k, from 2 to 9. Since when the number of cluster is agnostic, we may choose k base on the classification performance, to be precise, the sum of squared distances of samples to their closest cluster centre. The elbow plot with the classification objective plotted on the y-axis is presented in Fig 11.

As we can see, there is a sudden drop at $k = 3$, which indicates that the improvement of the objective slows down after $k = 3$, and therefore by trade-off between precision and efficiency, we may choose $k = 3$ if the true number of cluster is unknown.

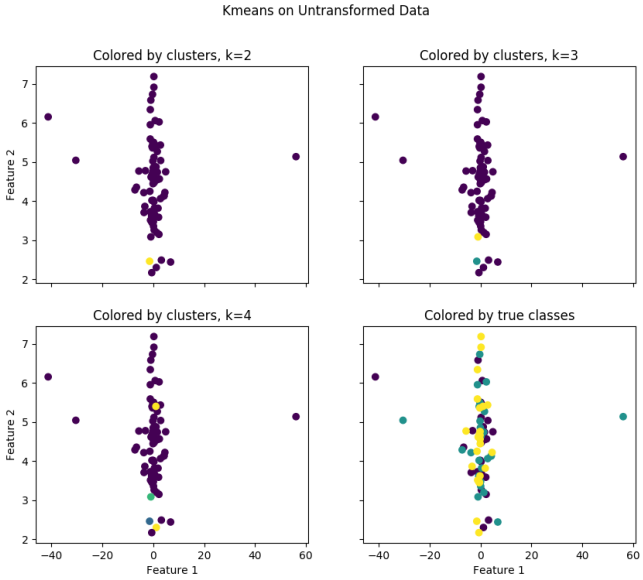


Figure 6: Classification result obtained from k-means algorithm, with $k = 2$, $k = 3$, $k = 4$.


Figure 7: Classification result obtained from k-means algorithm, with $k = 3$ on PCA transformed data.

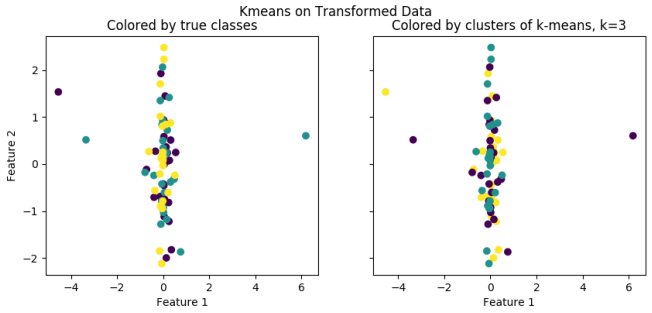


Figure 8: Classification result obtained from k-means algorithm, with $k = 3$ on scaled data.

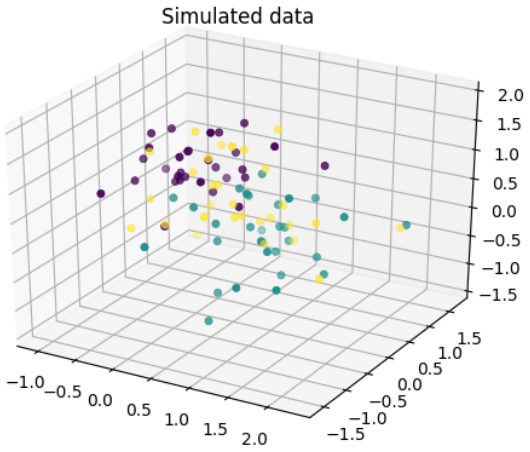


Figure 9: Simulated Data

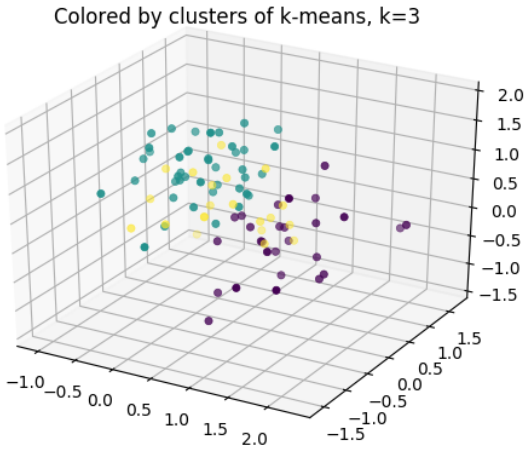


Figure 10: Classification with Kmeans, k=3.

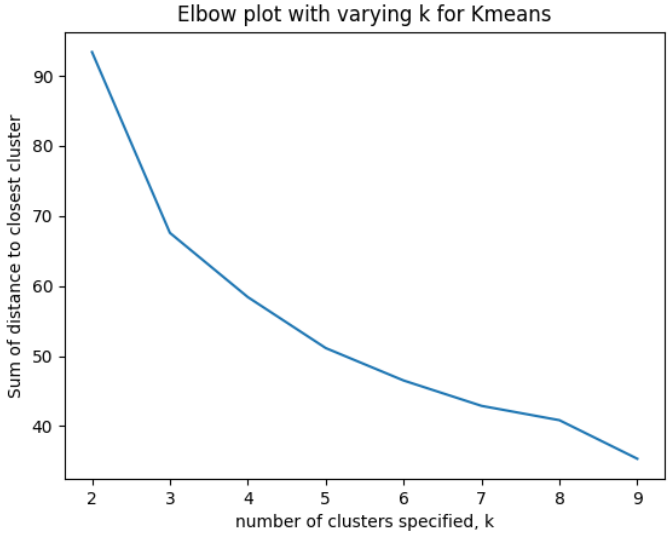


Figure 11: Classification with Kmeans, k=2,$\cdots$,9.