MACHINE LEARNING LAB

K-Means Clustering

MUNADI SIAL

SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

Features and Labels

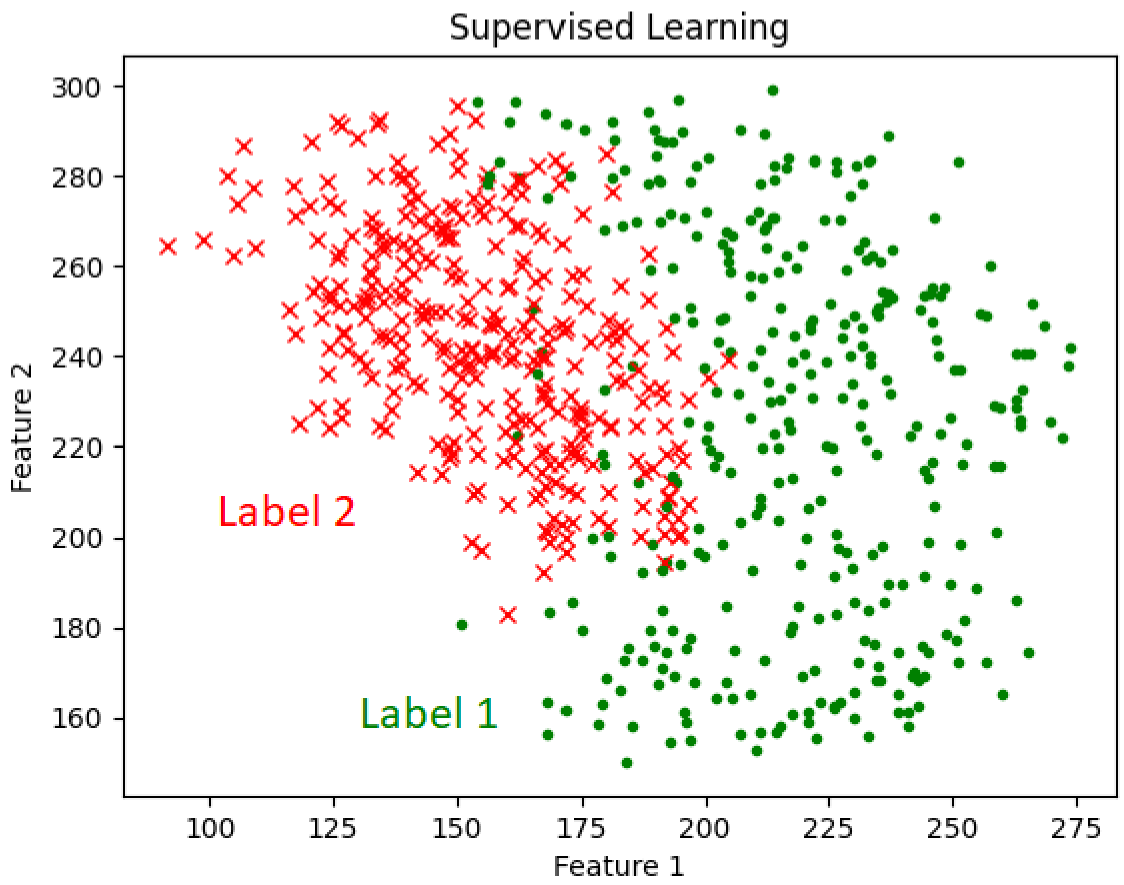
- Machine Learning consists of a wide variety of techniques that can be classified as Supervised, Unsupervised and Reinforcement Learning etc.

- A machine learning dataset essentially contains:
  - **Features:** the input columns in the dataset
  - **Labels:** the output columns in the dataset

- After the model is trained on the dataset, it is used to make a prediction (inference)
- To make a prediction, the user provides some new values for the features. These values are not from the dataset and are used by the trained model to give the output (prediction)

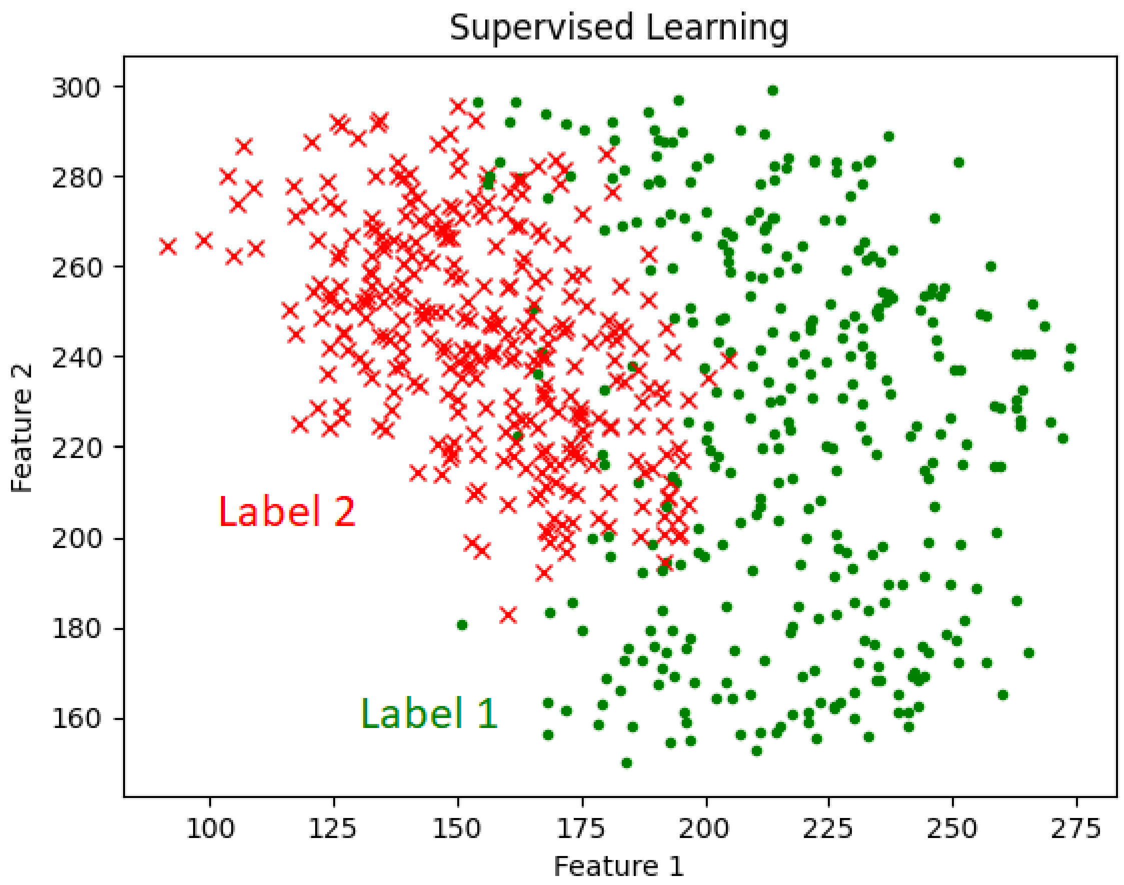
Supervised Learning

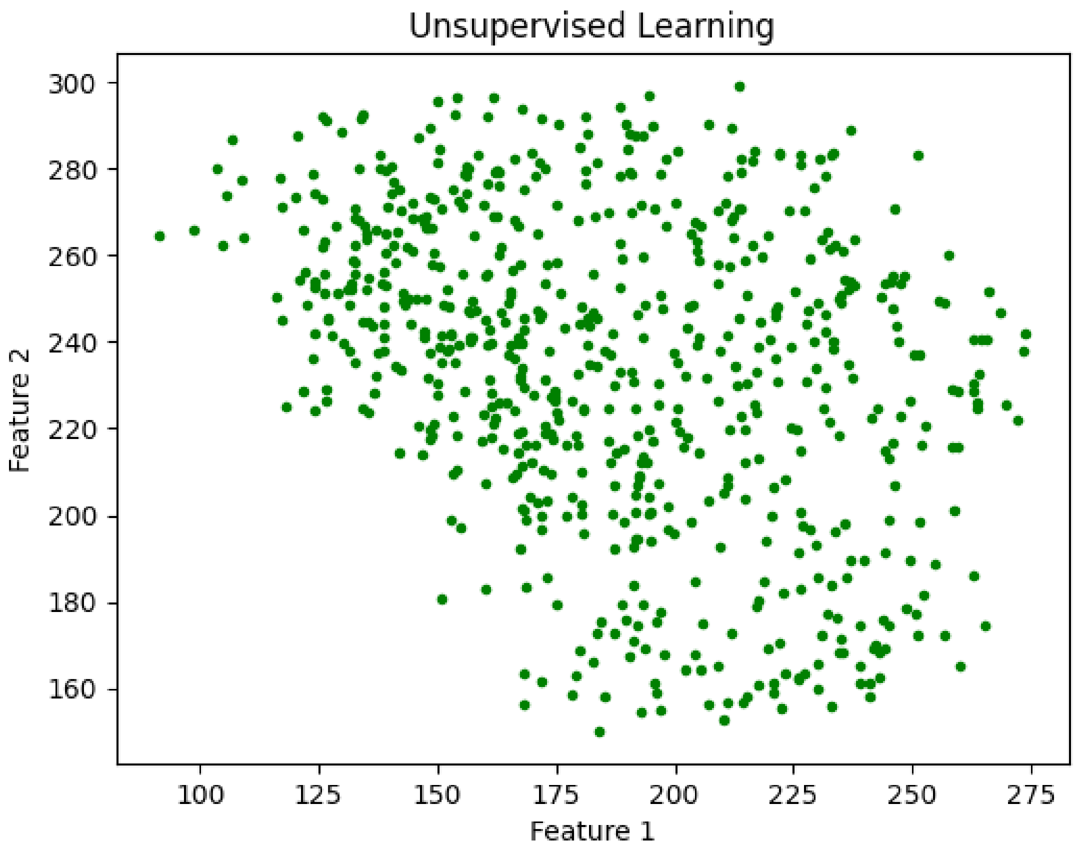
- Supervised Learning involves datasets that have both features and labels
- Examples include Linear Regression, Logistic Regression, Support Vector Machines, Neural Networks etc.



Unsupervised Learning

- Unsupervised Learning involves datasets with only features (no labels)
- Examples include K-means Clustering, Anomaly Detection and Dimensionality Reduction



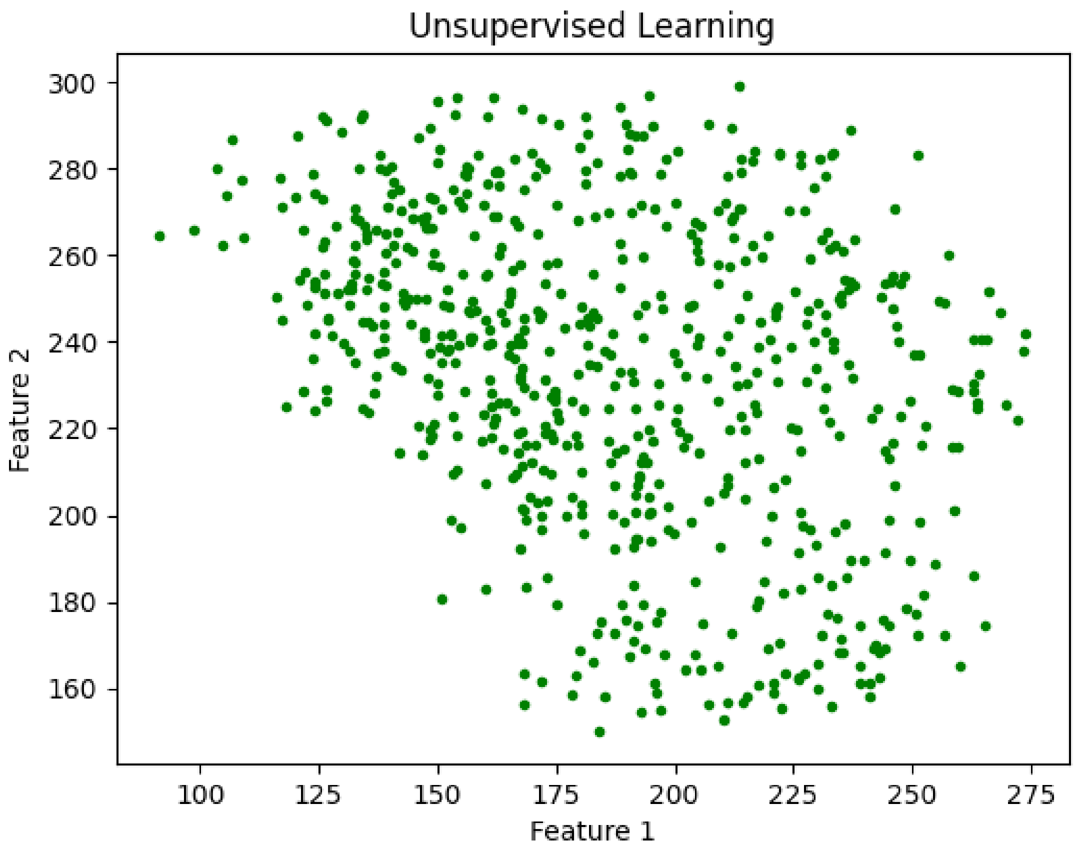


Unsupervised Learning

- In unsupervised learning, the goal is to find patterns, groups, similarities and structure from the distribution of the dataset features

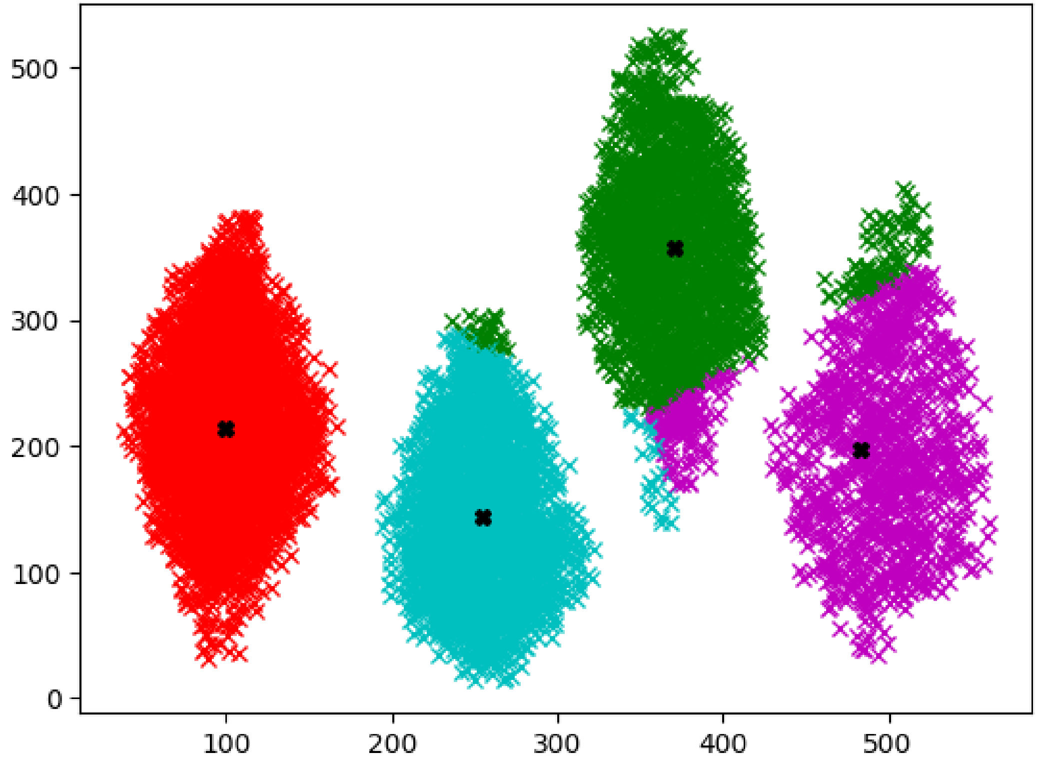
- In this lab, the focus will be solely on K-Means Clustering

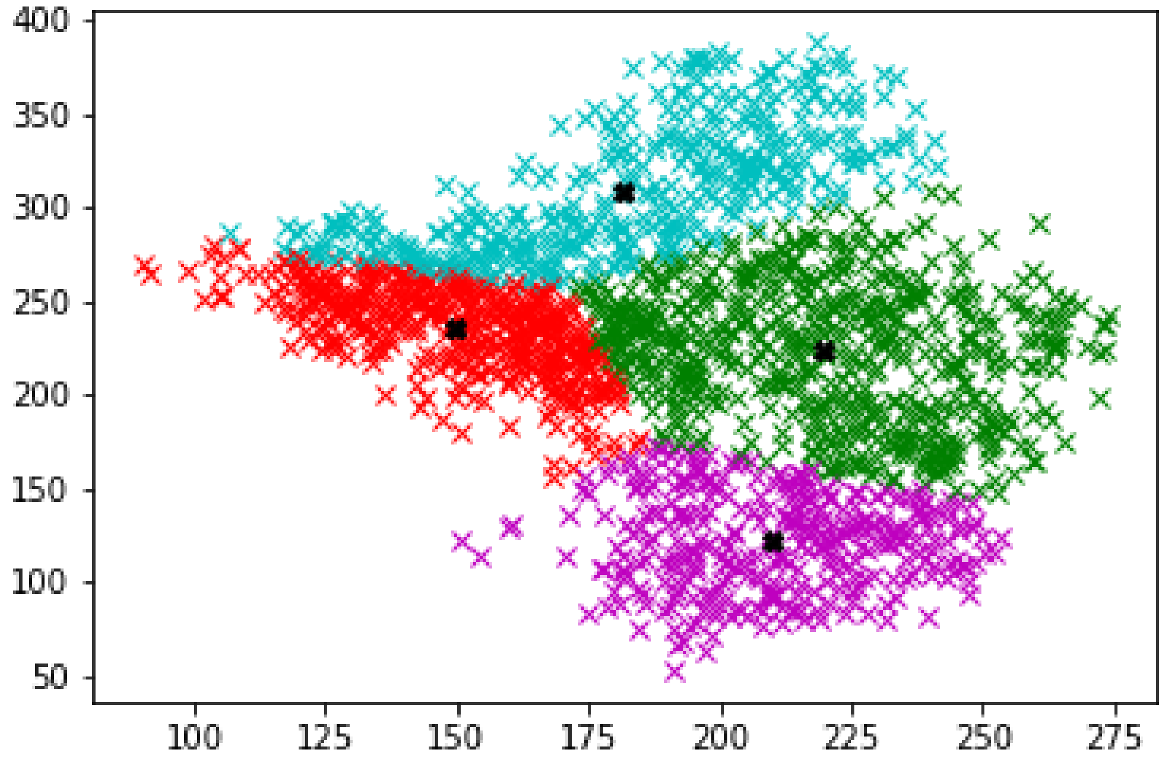
- Clustering is used widely in
  - Search Engines
  - Market Segmentation
  - Social Network Analysis
  - Astronomical Data Analysis
  - Image Segmentation



K-Means Clustering

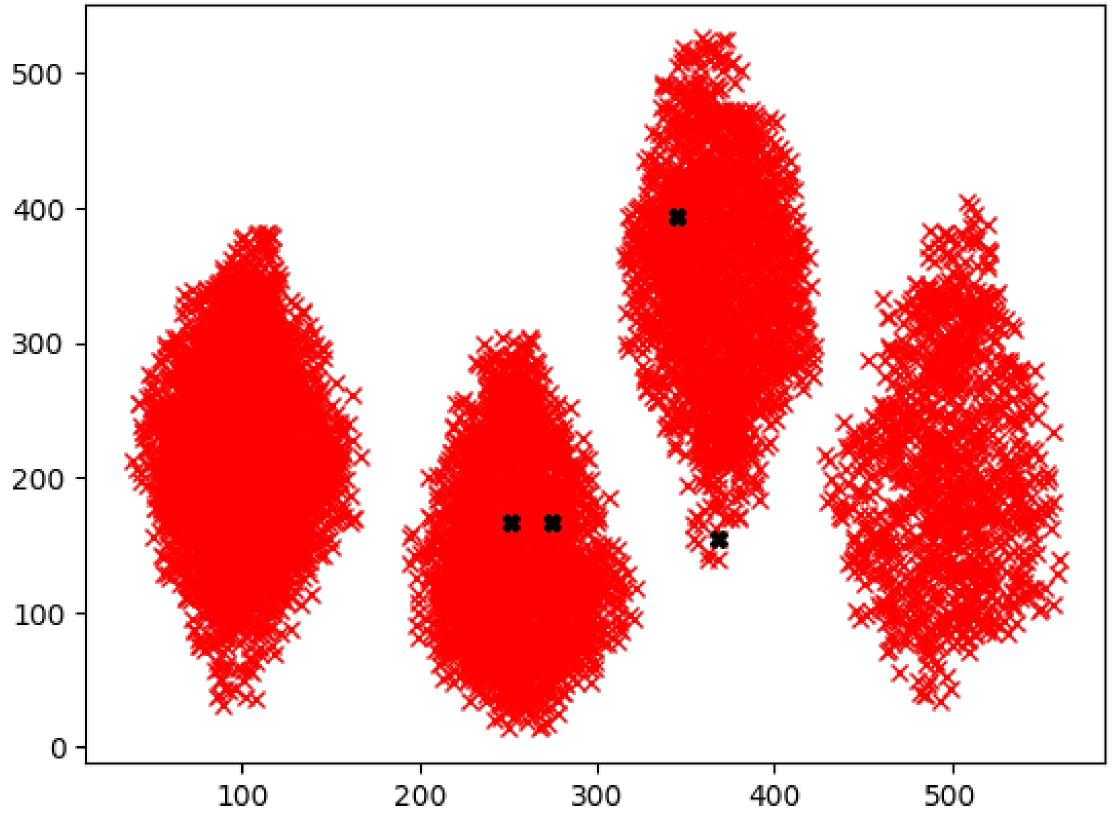
- K-Means Clustering is a technique that is used to segment the dataset into various groupings known as "clusters"
- The examples below show clustering in which each example point has a certain color showing its cluster. Black dots show the cluster *centroids*





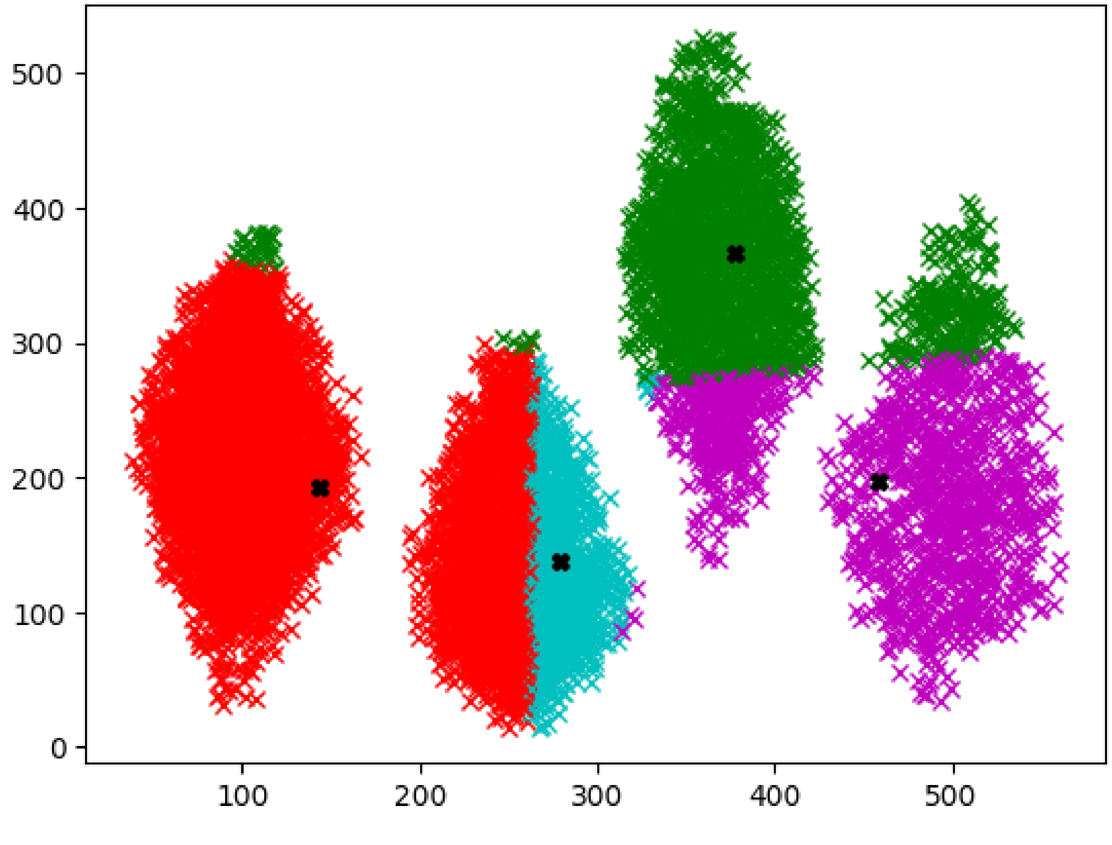
K-Means Clustering

- Before clustering starts, the programmer chooses the number of clusters denoted by K
- In the example below, K = 4 is chosen; 4 centroids are randomly obtained



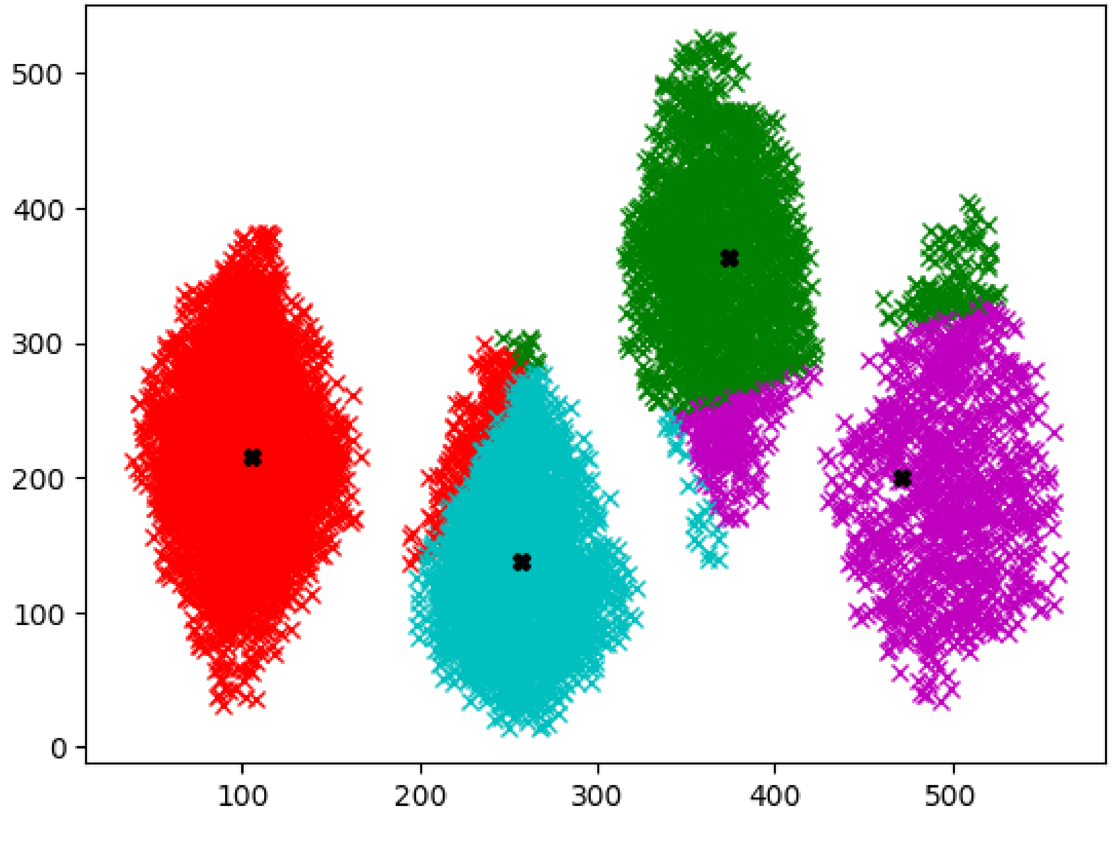
K-Means Clustering

- The centroids are iteratively updated using the dataset distribution



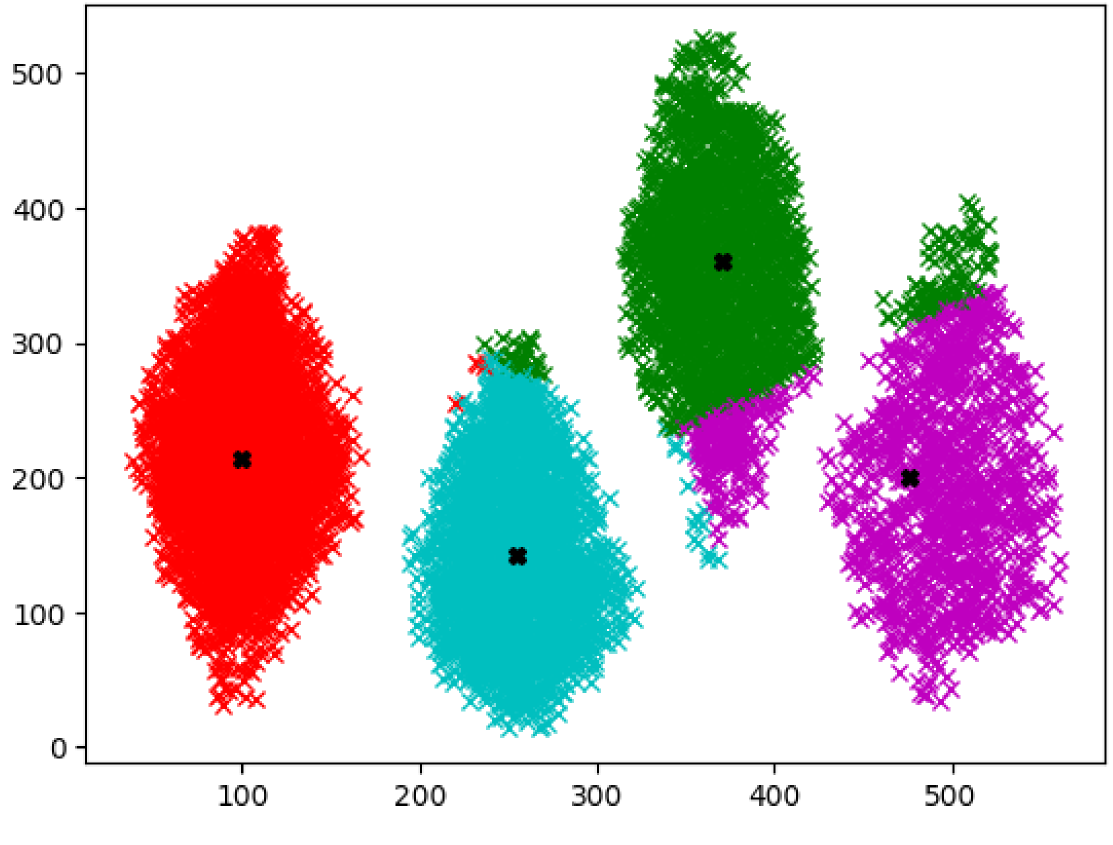
K-Means Clustering

- The centroids are iteratively updated using the dataset distribution



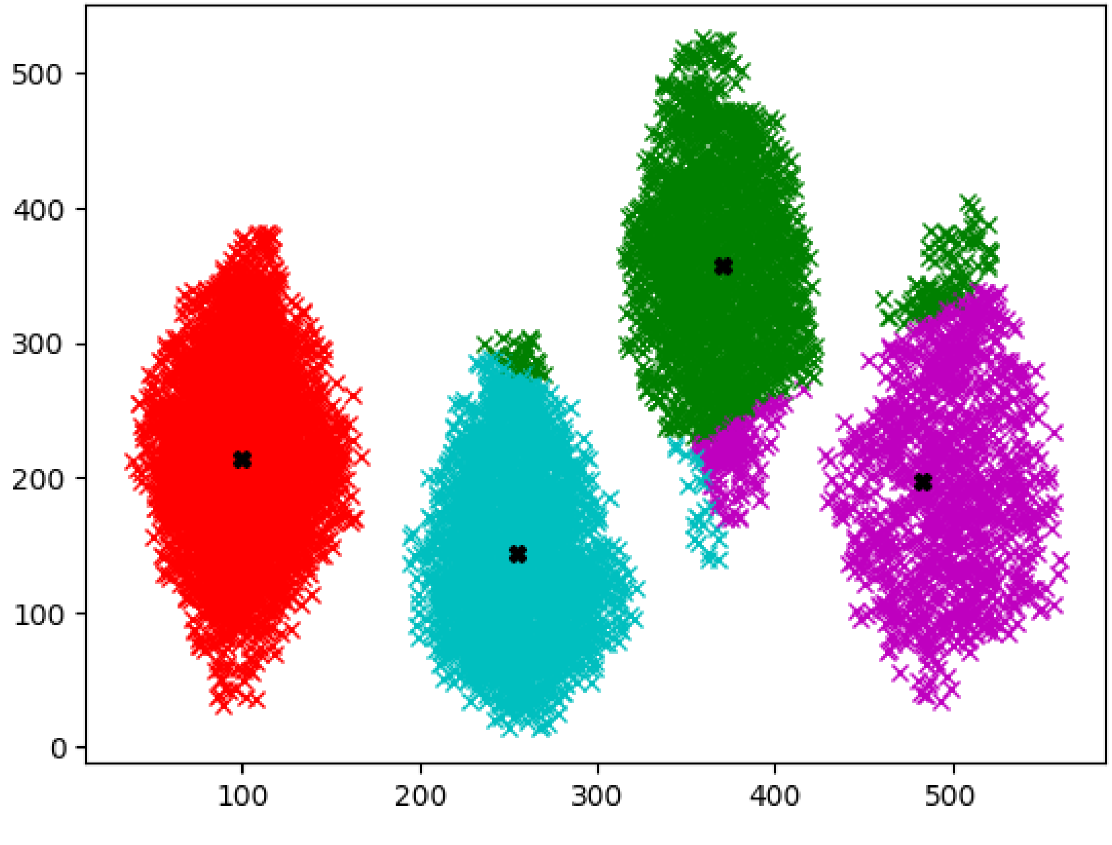
K-Means Clustering

- The centroids are iteratively updated using the dataset distribution

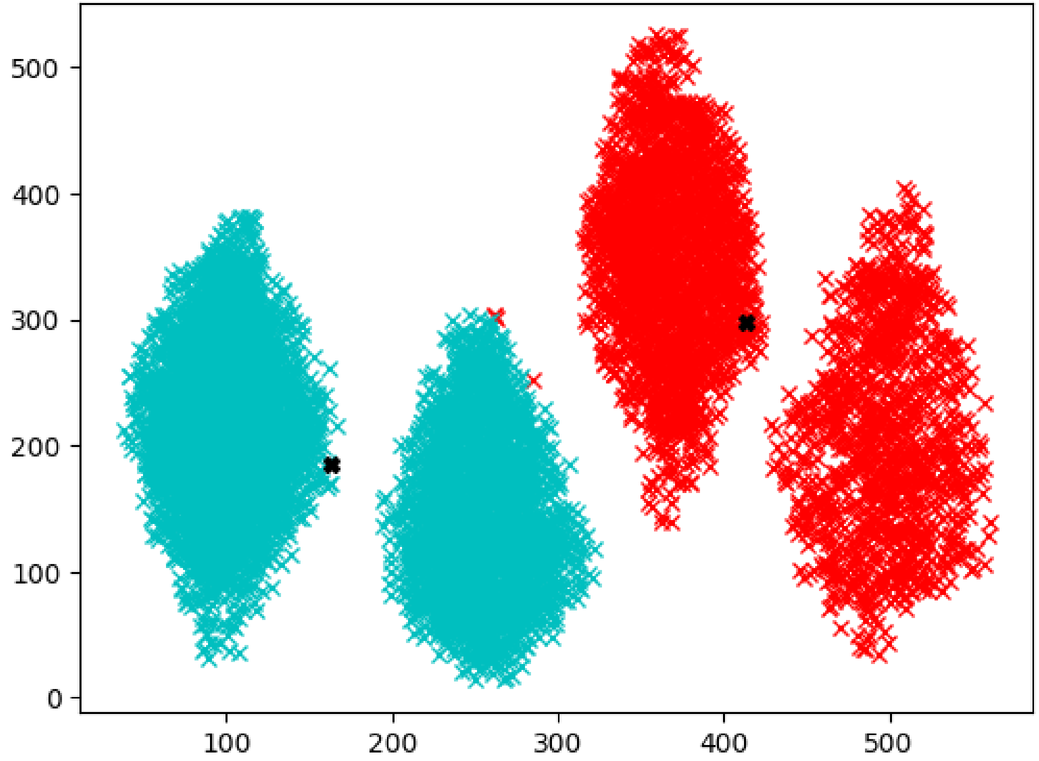


K-Means Clustering

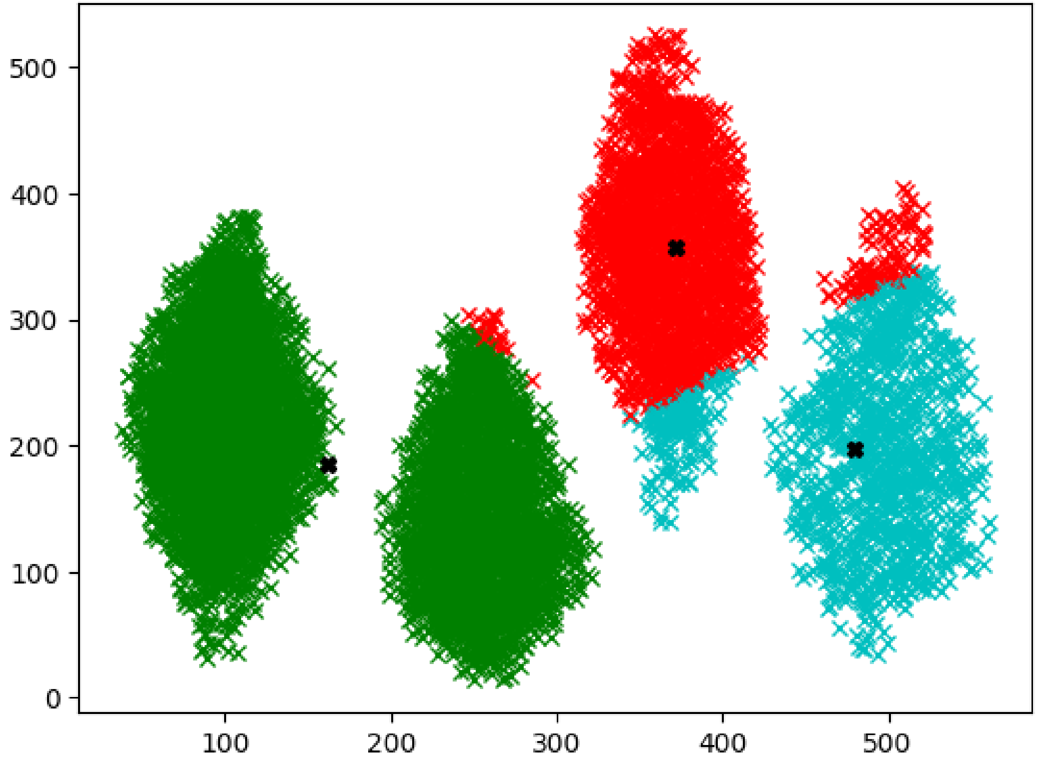
- The centroids are iteratively updated using the dataset distribution
- The number of iterations (epochs) is also to be chosen by the programmer



Number of Clusters

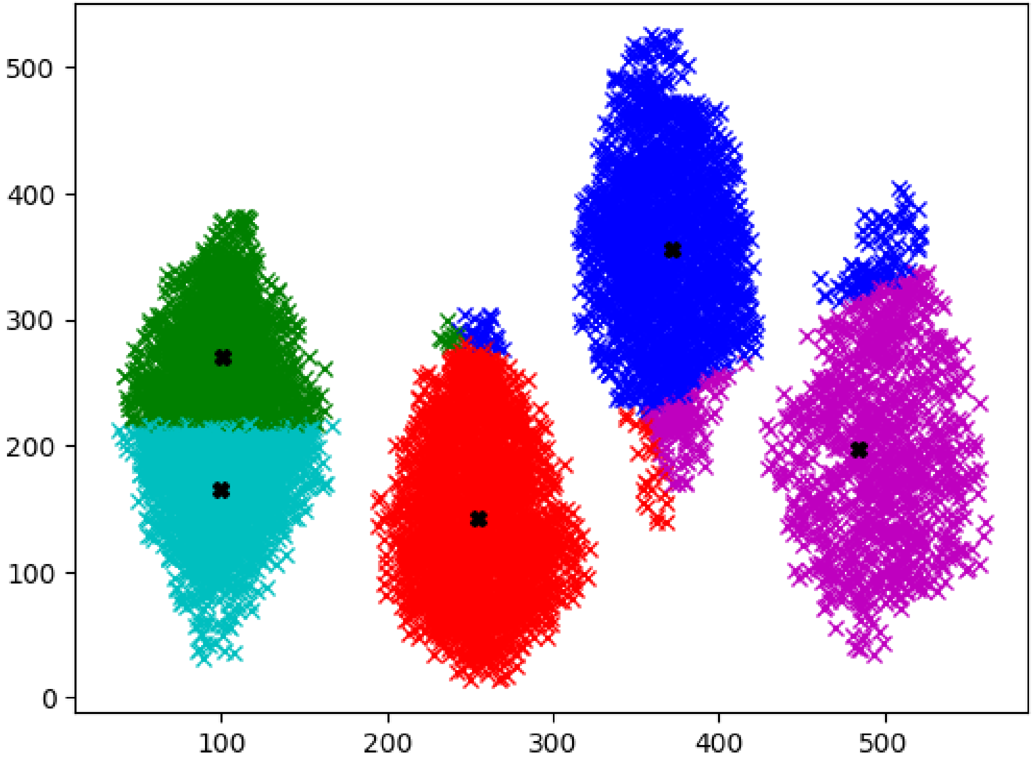


K = 2

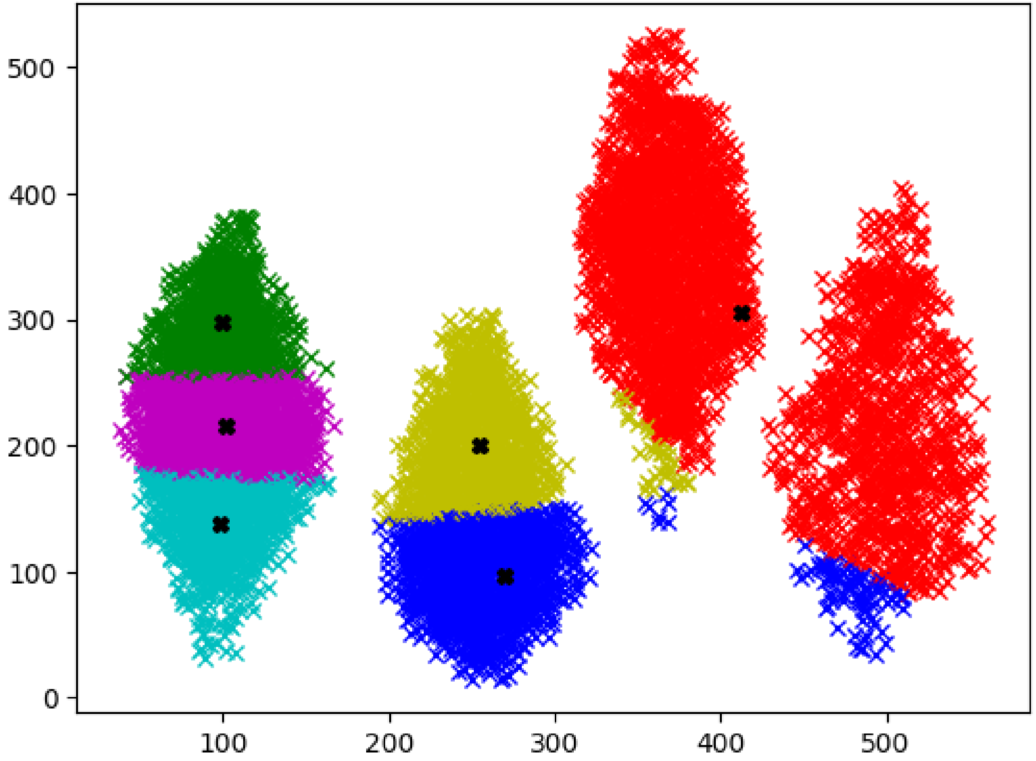


K = 3

Number of Clusters

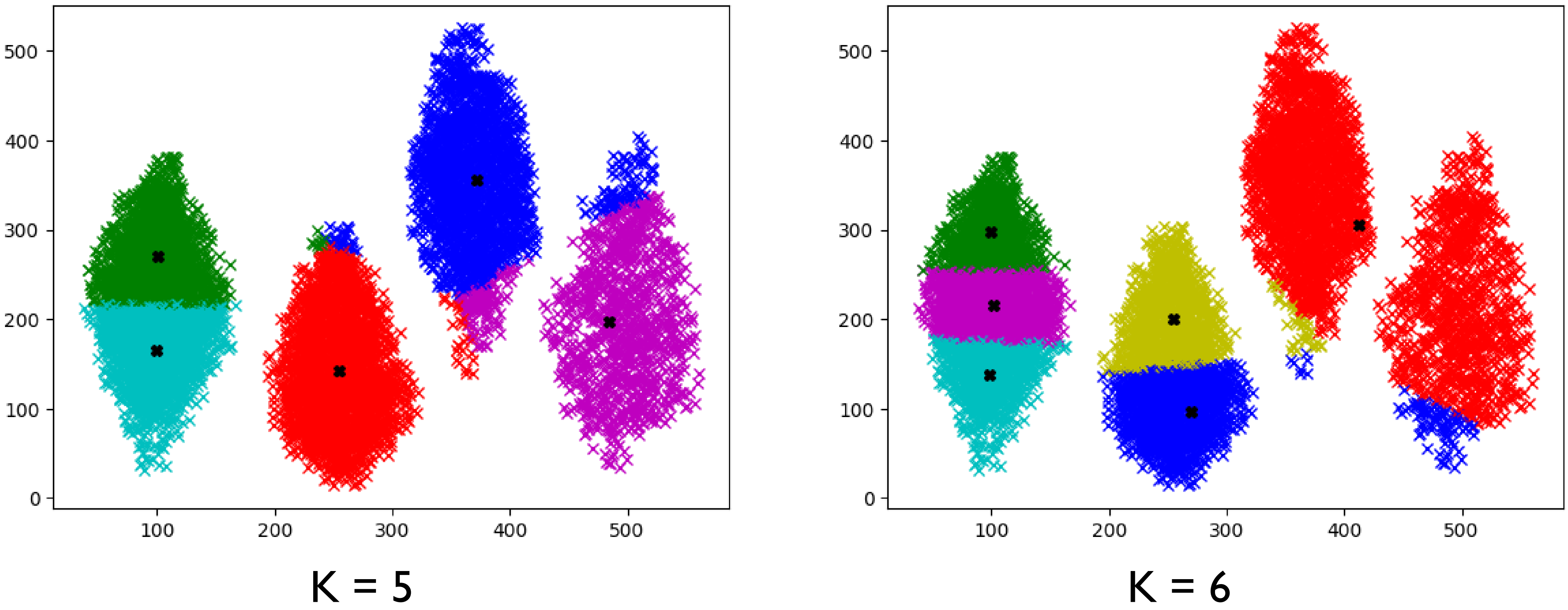


K = 5



K = 6

Number of Clusters



- The best value of K may be found by the “elbow method”

Clustering Algorithm

- The pseudocode for K-Means Clustering algorithm is given below:

```
specify K number of centroids
randomly initialize K number of centroids u
for j = 1:epochs
    for i = 1:m
       c(i) = index of closest cluster to training example
    for k = 1:K
       u(k) = mean of all training examples indexed to k
    plot of x1 and x2 clusters
```

- Before the clustering, you will need to load the dataset X
- You will also need to specify the epochs number

Clustering Algorithm

- The pseudocode for K-Means Clustering algorithm is given below:

```
specify K number of centroids
randomly initialize K number of centroids u
for j = 1:epochs
    for i = 1:m
       c(i) = index of closest cluster to training example
    for k = 1:K
       u(k) = mean of all training examples indexed to k
    plot of x1 and x2 clusters
```

- You can assign the initial centroids u_K using random K examples from the dataset

Clustering Algorithm

- The pseudocode for K-Means Clustering algorithm is given below:

```
specify K number of centroids
randomly initialize K number of centroids u
for j = 1:epochs
    for i = 1:m
       c(i) = index of closest cluster to training example
    for k = 1:K
       u(k) = mean of all training examples indexed to k
    plot of x1 and x2 clusters
```

- The main loop that will run the specified number of epochs
- Initially, start with smaller number of epochs to get an idea of how the clustering is proceeding. Opt for larger values later if needed

Clustering Algorithm

- The pseudocode for K-Means Clustering algorithm is given below:

```
specify K number of centroids
randomly initialize K number of centroids u
for j = 1:epochs
    for i = 1:m
        c(i) = index of closest cluster to training example
    for k = 1:K
        u(k) = mean of all training examples indexed to k
    plot of x1 and x2 clusters
```

m : number of training examples in the dataset

c(i) : vector used to store indices and has length equal to m

Clustering Algorithm

- The pseudocode for K-Means Clustering algorithm is given below:

```
specify K number of centroids
randomly initialize K number of centroids u
for j = 1:epochs
   for i = 1:m
      c(i) = index of closest cluster to training example
   for k = 1:K
      u(k) = mean of all training examples indexed to k
   plot of x1 and x2 clusters
```

- In this loop, you go through each training example one-by-one and determine which of the centroids is closest to each example
- The index of the closest centroid (1, 2, 3 … K) is stored in the c(i) vector

Clustering Algorithm

- The pseudocode for K-Means Clustering algorithm is given below:

```
specify K number of centroids
randomly initialize K number of centroids u
for j = 1:epochs
    for i = 1:m
        c(i) = index of closest cluster to training example
    for k = 1:K
        u(k) = mean of all training examples indexed to k
    plot of x1 and x2 clusters
```

To determine the closest centroid, the Euclidean distance of the example from the centroid is computed:

$$d = \sqrt{(x_1 - u_{1,K})^2 + (x_2 - u_{2,K})^2 + (x_3 - u_{3,K})^2}$$

Clustering Algorithm

- The pseudocode for K-Means Clustering algorithm is given below:

```
specify K number of centroids
randomly initialize K number of centroids u
for j = 1:epochs
    for i = 1:m
       c(i) = index of closest cluster to training example
    for k = 1:K
       u(k) = mean of all training examples indexed to k
    plot of x1 and x2 clusters
```

- After completing the c(i) vector, it is used to update the centroid positions
- Notice this loop iterates through each of the K clusters
- The training examples that belong to a particular cluster are averaged to give the new centroid point for that cluster

Clustering Algorithm

- The pseudocode for K-Means Clustering algorithm is given below:

```
specify K number of centroids
randomly initialize K number of centroids u
for j = 1:epochs
    for i = 1:m
       c(i) = index of closest cluster to training example
    for k = 1:K
       u(k) = mean of all training examples indexed to k
    plot of x1 and x2 clusters
```

- At the end of each iteration (epoch), it is a good idea to plot the clusters
- By analyzing the plots at each epoch, we can get a sense of how the algorithm is proceeding and decide if we should change the epoch number and value of K

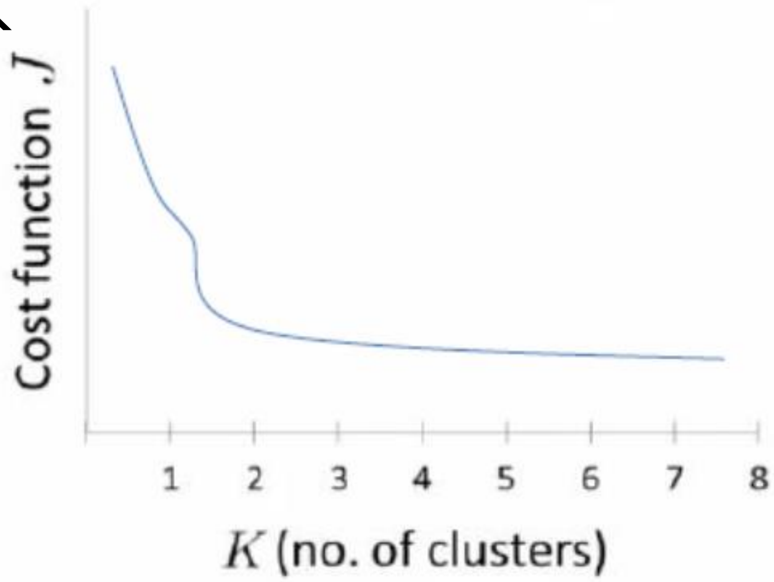
Choosing K Value

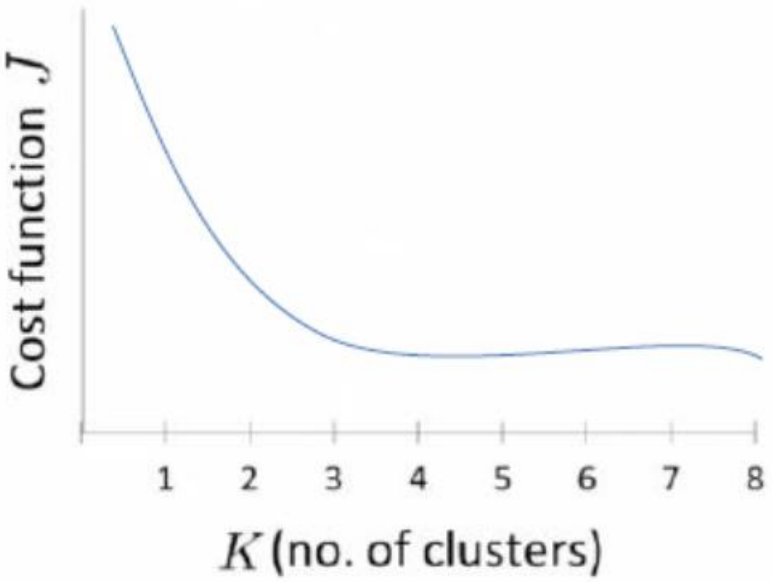
- At times, the best value of K may not be obvious
- To get the best value of K, we can run the clustering algorithm for different values of K and for each K value, a cost can be computed:

$$Cost_K = J(K) = \frac{1}{m}\sum_{i=1}^{m} ||(X^{(i)} - u_K^{(i)})||^2$$

- A plot of the cost for each K value can be obtained
- By looking at where the "elbow" is, the optimum value of K can be found

- Note that this method does not always work





Lab Tasks

- Download the materials from LMS
- Perform the Lab Tasks given in the manual
- Convert the completed manual into .pdf and submit on LMS