

**Guided Projects: Unsupervised Learning**  
**K-Means Clustering: Image Segmentation**

<b>Name</b>	<b>Lakshmi Thirunavukkarasu</b>
<b>Course</b>	<b>AI and ML (Batch 5)</b>
<b>Problem Statement</b>	<b>Implement image segmentation using K-Means.</b>

### **Software requirements perquisites**

1. Anaconda
2. Python 3.8
3. Python Packages
  - Numpy
  - Matplotlib

### **Steps**

1. Create a K\_Means Class object with functions listed below:

**cluster\_initialization** : To initialize the cluster labels

**calculate\_mean** : Calculate mean for each cluster

**calculate\_distance** : Calculate the Euclidean distance from the cluster centroids

**update\_label** : Update the label for each data point based on minimum distance from the cluster centroid

**fit** : Model that fits the data points into clusters.

## 2. Instantiate K\_Means object to cluster the IRIS Dataset

### Call the K\_Means class for the IRIS Dataset

```
In [36]: df1 = pd.read_csv("IRIS.csv")
df = np.matrix(df1.drop('species', axis = 1))
kmeans = K_Means(no_of_clusters=3, max_iter=10, threshold = .001)
mean, labels = kmeans.fit(df)
print(mean)
print(labels)
print(type(np.array(labels)))
```

## 3. Instantiate K\_Means object for Image segmentation for the fruits image.

```
In [69]: #Convert the image dataset into 2d array
x, y, z = image.shape
image_2d = image.reshape(x*y, z)
print(image_2d.shape)
print(image_2d[:2,:])
```

```
(50325, 3)
[[ 7 17  9]
 [ 9 16  9]]
```

```
In [56]: #Call the K-Means class
kmeans_image = K_Means(no_of_clusters=5, max_iter=10, threshold = .001)
mean, labels = kmeans_image.fit(image_2d)
print(mean)
```

## 4. Again apply K\_Means classification for the IRIS flowers.

```
In [44]: kmeans_iris = KMeans(no_of_clusters=5, max_iter=10, threshold = .001)
mean, labels = kmeans_iris.fit(image_2d)
print(mean)

[[ 45.52901826  59.88085859  59.38568921]
 [ 84.89691874 101.43763698  93.89809337]
 [134.91618116 156.92015806 127.90878464]
 [204.02312095 206.19717382 243.35306929]
 [128.28512092 112.68756394 199.92431559]]
```

```
In [45]: plt.figure(figsize = (15,8))
plt.imshow((mean[labels].reshape(x, y)).astype('uint8'))
```

```
Out[45]: <matplotlib.image.AxesImage at 0x18628987d00>
```

