Assignment Project Report
K-Means Clustering: Image Segmentation

Name: Pavan Tiwari

Course: AI and ML
(Batch 4)

- **Problem Statement**

Take a bright colorful image (Eg: image having fruits in it) and implement image segmentation using K-Means. You can first try to implement K-Means on iris dataset to understand its working and then extend the same logic, using the image pixels as the data points. Hint: All the K centroids will represent a color and therefore, you can initialize all the pixels to belong to a cluster randomly and then start the training of the centroids.

- Software:
  - Python 3 (Use anaconda as your python distributor as well)

- Tools:
  - Pandas
  - Numpy
  - Matplotlib
  - Seaborn
  - OpenCv

- Dataset Link:
  https://www.kaggle.com/teejmahal20/airline-passenger-satisfaction

- **Method Used**

Image segmentation is the process of partitioning a digital image into multiple distinct regions containing each pixel with similar attributes i.e. classification of an image into different groups. There are different methods, and one of the most popular methods is the k-means clustering algorithm. K-Means clustering algorithm is an unsupervised algorithm, and it is used to segment the interest area from the background.

- **Implementation:**

1. Load all required libraries

In [1]:
```
# Importing the necessary Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn
from sklearn.datasets import load_iris
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
```

In [2]:
```
#Loading the Data
iris = load_iris()
```

2. Visualizing data

EDA ¶

In [4]:
```
data.feature_names
```

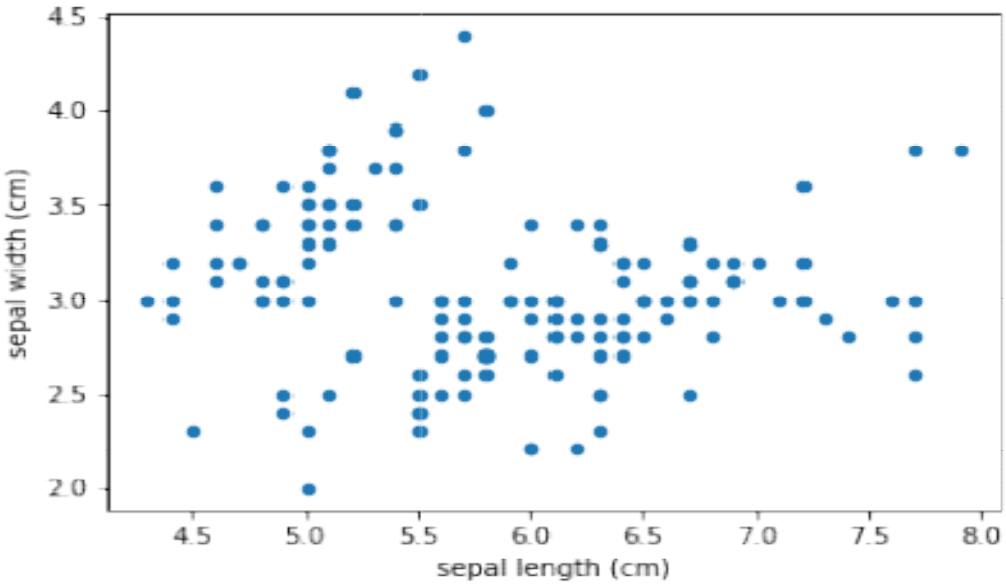
Out[4]:
```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

In [10]:
```
df = pd. DataFrame(data= np. c_[iris['data'], iris['target']. astype('int32')],
                   columns= iris['feature_names'] + ['target'])
```

In [11]:
```
df.head()
```

Out[11]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0.0
1	4.9	3.0	1.4	0.2	0.0
2	4.7	3.2	1.3	0.2	0.0
3	4.6	3.1	1.5	0.2	0.0
4	5.0	3.6	1.4	0.2	0.0



3. Model Building Using KMeans On Iris Dataset

```
kmeans = KMeans(n_clusters=3,init = 'k-means++',   max_iter = 100, n_init = 10, random_state = 0)
```

```
#Predicting the cluster for our data
y_kmeans = kmeans.fit_predict(x)
```

```
print(kmeans.cluster_centers_)
```

```
.9016129  2.7483871  4.39354839 1.43387097]
.006      3.428      1.462      0.246     ]
.85       3.07368421 5.74210526 2.07105263]]
```

```
y_kmeans
```

```
ay([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
    1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
    1, 1, 1, 1, 1, 1, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 2, 2, 2, 2, 0, 2, 2, 2,
    2, 2, 2, 0, 0, 2, 2, 2, 2, 0, 2, 0, 2, 0, 2, 2, 0, 0, 2, 2, 2, 2,
    2, 0, 2, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 0])
```

```
#Visualising the clusters
```

4. Image Segmentation Using K Means

```
import cv2
import os
import matplotlib.image as mpimg
```

```
# Reading the image
img = cv2.imread("fruits.jpg")
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

```
plt.figure(figsize=(10,10))
plt.imshow(img)
```

5. Output Of image Segmentation