**You need to implement K-mean Clustering from Scratch , on IRIS Data set.**

## About Data-Set

The **Iris Dataset** is multivariate flowers datset contains four features (length and width of sepals and petals) of 50 samples of three species of **Iris** (**Iris** setosa, **Iris** virginica and **Iris** versicolor) with the total of 150 samples.

```
subset(iris, Species == "setosa")[1:5,]
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
```

```
subset(iris, Species == "versicolor")[1:5,]
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 51          7.0         3.2          4.7         1.4 versicolor
## 52          6.4         3.2          4.5         1.5 versicolor
## 53          6.9         3.1          4.9         1.5 versicolor
## 54          5.5         2.3          4.0         1.3 versicolor
## 55          6.5         2.8          4.6         1.5 versicolor
```

```
subset(iris, Species == "virginica")[1:5,]
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 101          6.3         3.3          6.0         2.5 virginica
## 102          5.8         2.7          5.1         1.9 virginica
## 103          7.1         3.0          5.9         2.1 virginica
## 104          6.3         2.9          5.6         1.8 virginica
## 105          6.5         3.0          5.8         2.2 virginica
```

You need to Implement 2 most important steps of Clustering.
1- **Cluster Assignment** : Assign each data value to its closest possible cluster.
2- **Computing Centroid:** Update Clusters by taking mean of each clusters.

**Cluster assignment algorithm**

```
# Assign every training example x^(i) to its closest centroid, given the current centroid positions
def find_closest_centroids(X, centroids):
    '''returns the array of assigned clusters to each example '''
    m = X.shape[0] # no of datappints
    k = centroids.shape[0]  #division size
    idx = np.zeros(m) # array to assign the centriod

    # Your Code Here

    return idx
```

**Computing centoroid means**

```
# Update Centroids by taking mean of each centroid values
def compute_centroids(X, idx, k):
    m, n = X.shape
    centroids = np.zeros((k, n))  #3*4
    '''Return Updated Values of all K centroids'''

    # Your Code Here

    return centroids
```