LAB CYCLE 7

Experiment No : 7
Date : 28/02/2022

Aim:

Write a program to cluster a set of points using K-means. Consider K=3 as the number of clusters. Use Euclidean distance as the distance measure. Randomlyinitialize a cluster mean as one of the data points. Iterate for 10 iterations.
After iterations are over, print the final cluster means for each of the clusters. Use the ground truth cluster label present in the data set to compute and printthe Jacquard distance of the obtained clusters with the ground truth clusters for each of the three clusters.

Data Set Preparation:

Data Filename: data_iris.csv

The data set contains 150 data points, there are three clusters where eachcluster refers to a type of iris plant. The first four columns represent the attributes listed below.

Note that only the first four columns should be used as attributes. The last column is the ground truth cluster name and is to be used for evaluating thecluster quality.

1. **sepal length in cm**
2. **sepal width in cm**
3. **petal length in cm**
4. **petal width in cm**
5. **Ground truth cluster name:**

-- Iris Setosa

-- Iris Versicolour

-- Iris Virginica

Source Code:

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [3]:
```
datairis=pd.read_csv('data_iris.csv',header=None)
```

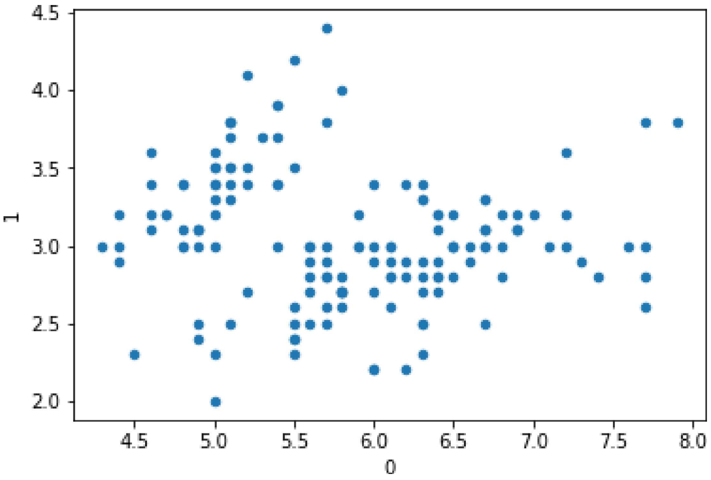
In [4]:
```
datairis.head()
```

Out[4]:
```
     0    1    2    3            4
0  5.1  3.5  1.4  0.2  Iris-setosa
1  4.9  3.0  1.4  0.2  Iris-setosa
2  4.7  3.2  1.3  0.2  Iris-setosa
3  4.6  3.1  1.5  0.2  Iris-setosa
4  5.0  3.6  1.4  0.2  Iris-setosa
```

In [5]:
```
datairis.describe()
```

Out[5]:
```
                0           1           2           3
count  150.000000  150.000000  150.000000  150.000000
mean     5.843333    3.054000    3.758667    1.198667
std      0.828066    0.433594    1.764420    0.763161
min      4.300000    2.000000    1.000000    0.100000
25%      5.100000    2.800000    1.600000    0.300000
50%      5.800000    3.000000    4.350000    1.300000
75%      6.400000    3.300000    5.100000    1.800000
max      7.900000    4.400000    6.900000    2.500000
```

In [12]:
```
datairis.plot(kind="scatter",x=0,y=1)
plt.show()
```



In [13]:

```
#Preprocessing the dataset :
```

In [14]:

```
from sklearn.model_selection import train_test_split
x = datairis.iloc[:, :-1].values #last column values excluded
y = datairis.iloc[:,   -1].values #last column value
from sklearn.model_selection  import  train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=0)
```

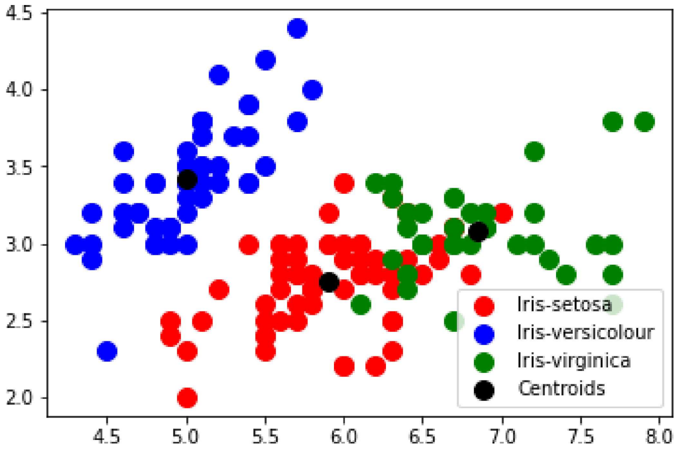
In [15]:

```
from sklearn.datasets import load_iris
from sklearn.cluster import KMeans
```

In [16]:

```
iris_data=load_iris()   #loading iris dataset from sklearn.datasets
iris_df = pd.DataFrame(iris_data.data, columns = iris_data.feature_names) #creating
kmeans = KMeans(n_clusters=3,init = 'k-means++',    max_iter = 100, n_init = 10, rand
y_kmeans = kmeans.fit_predict(x)
print(kmeans.cluster_centers_) #display cluster centers
plt.scatter(x[y_kmeans   == 0, 0], x[y_kmeans == 0, 1],s = 100, c = 'red', label = '
plt.scatter(x[y_kmeans   == 1, 0], x[y_kmeans == 1, 1],s = 100, c = 'blue', label =
plt.scatter(x[y_kmeans   == 2, 0], x[y_kmeans == 2, 1],s = 100, c = 'green', label =
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:,1],s = 100, c
plt.legend()
plt.show()
```

```
[[5.9016129  2.7483871  4.39354839 1.43387097]
 [5.006      3.418      1.464      0.244     ]
 [6.85       3.07368421 5.74210526 2.07105263]]
```


4.5
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
Iris-setosa
Iris-versicolour
Iris-virginica
Centroids