PRACTICAL – 9

Implementation of K Mean Clustering and un-clustering on jupyter Notebook using Python.

Step 1

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
iris = sns.load_dataset('iris')
labels = iris.species.unique()
iris.head()
```

output

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Step 2

```
iris["species"] = pd.Categorical(iris["species"])
iris["species"] = iris["species"].cat.codes
iris.head()
```

Output

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

Step 3

```
X = iris[['sepal_length','sepal_width']].values
y = iris.species
from sklearn.cluster import KMeans
model = KMeans(n_clusters = 3).fit(X)
centers = model.cluster_centers_
new_labels = model.labels_
print('Centroids :',centers)
```

output

```
Centroids : [[5.006      3.428     ]
 [6.81276596 3.07446809]
 [5.77358491 2.69245283]]
```

Step 4

```
print('\nLabels :',new_labels)
```

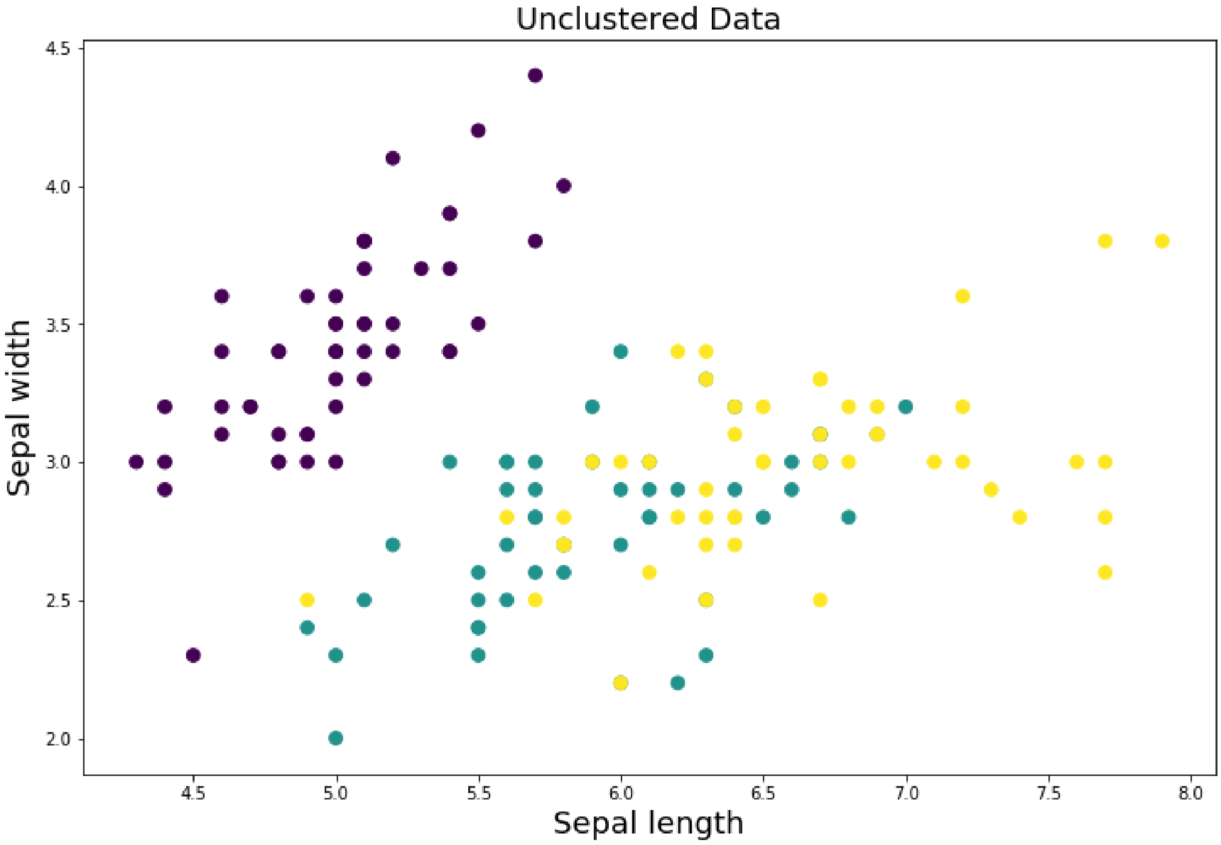
output

```
Labels : [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 2 1 2 1 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2
 1 1 1 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 2 1 1 1 1
 1 1 2 2 1 1 1 1 2 1 2 1 2 1 1 2 2 1 1 1 1 1 2 2 1 1 1 2 1 1 1 2 1 1 1 2 1
 1 2]
```

Step 5

```
plt.figure(figsize=(12,8))
plt.scatter(X[:, 0], X[:, 1],c=y, s=60)
plt.xlabel('Sepal length', fontsize=18)
plt.ylabel('Sepal width', fontsize=18)
plt.title('Unclustered Data',fontsize=18)
```

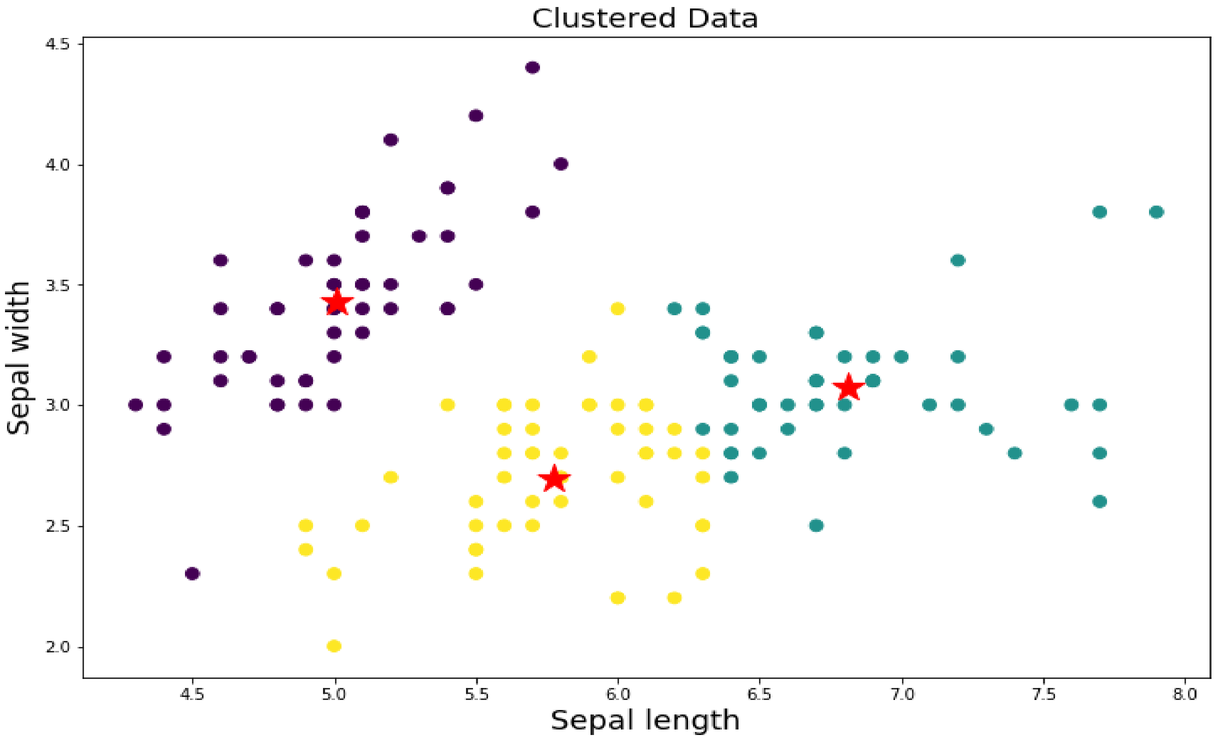
output



Step 6

```
plt.figure(figsize=(12,8))
plt.scatter(X[:, 0], X[:, 1], c=new_labels,s=60)
plt.scatter(centers[:, 0], centers[:, 1], c='r', s=400, marker = '*', zorder=10);
plt.xlabel('Sepal length', fontsize=18)
plt.ylabel('Sepal width', fontsize=18)
plt.title('Clustered Data',fontsize=18)
```

output



Step 7

```
y_pred = model.predict([[2.3,5.6]])
print("Result :",labels[y_pred[0]])
```

Result : setosa