Lab 9

January 19, 2018

1 Lab 9

1.1 *Part A*

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

```
In [2]: from sklearn.datasets.samples_generator import make_blobs
```

1.1.1 Generate synthetic two-dimenstional data

```
In [3]: x, y_true = make_blobs(n_samples=300, centers=4, cluster_std=0.64, random_state=0)
```

1.1.2 Display X & y_true. What is the size?

```
In [5]: x.shape
```

```
Out[5]: (300, 2)
```

```
In [6]: y_true.shape
```

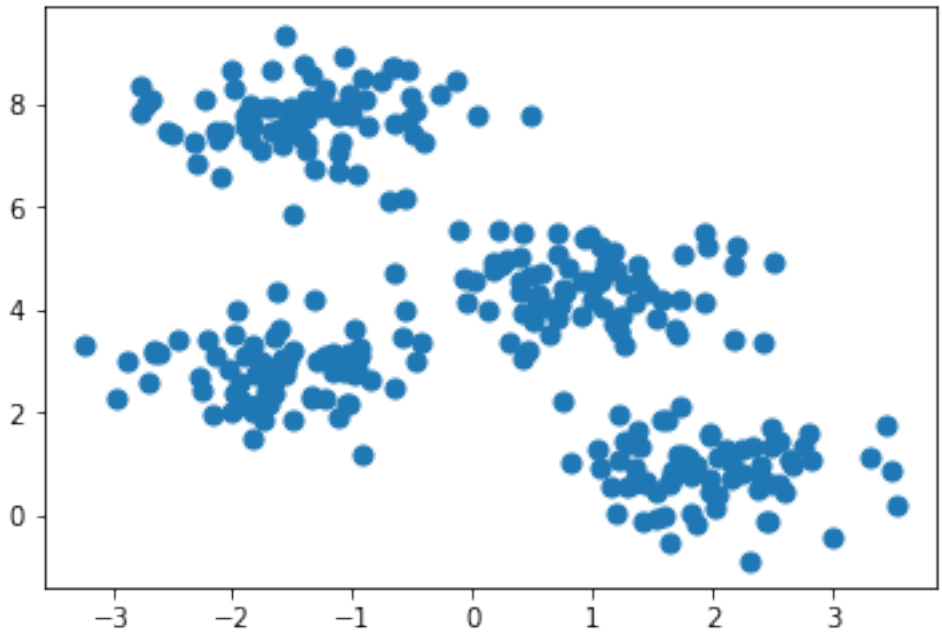
```
Out[6]: (300,)
```

1.1.3 Visualize the data using scatter plot

```
In [7]: plt.scatter(x[:,0], x[:,1], s=50)
```

```
Out[7]: <matplotlib.collections.PathCollection at 0x7fd577564518>
```

```
In [9]: plt.show()
```




```
In [10]: from sklearn.cluster import KMeans
```

1.1.4 Build the clustering model

```
In [11]: kmeans = KMeans(n_clusters=4)
```

1.1.5 Train the data

```
In [12]: kmeans.fit(x)
```

```
Out[12]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
             n_clusters=4, n_init=10, n_jobs=1, precompute_distances='auto',
             random_state=None, tol=0.0001, verbose=0)
```

1.1.6 kmeans.labels_attribute

```
In [27]: kmeans.labels_
```

```
Out[27]: array([0, 1, 2, 1, 0, 0, 3, 2, 1, 1, 3, 1, 2, 1, 0, 2, 2, 0, 3, 3, 0, 0, 2,
                3, 3, 2, 0, 2, 3, 2, 1, 1, 2, 1, 1, 1, 1, 1, 3, 0, 2, 3, 2, 2, 3, 3,
                1, 3, 1, 0, 3, 0, 1, 0, 0, 3, 1, 3, 1, 0, 1, 2, 1, 3, 3, 3, 1, 0, 1,
                3, 2, 3, 1, 3, 3, 1, 3, 2, 0, 1, 0, 2, 0, 0, 1, 2, 0, 2, 1, 1, 2, 0,
                1, 3, 3, 2, 0, 0, 2, 3, 1, 0, 1, 0, 2, 0, 0, 2, 1, 2, 3, 3, 0, 1, 0,
                2, 1, 0, 0, 2, 3, 0, 3, 0, 0, 0, 0, 3, 0, 3, 1, 3, 3, 0, 1, 3, 3, 1,
                2, 1, 1, 3, 2, 3, 2, 3, 1, 2, 1, 1, 1, 2, 1, 2, 0, 3, 1, 3, 0, 2, 1,
                2, 2, 0, 2, 3, 3, 2, 0, 2, 2, 1, 0, 2, 3, 1, 0, 0, 2, 3, 0, 2, 3, 3,
```

```
2, 2, 2, 2, 0, 1, 2, 3, 2, 2, 3, 3, 3, 2, 3, 1, 2, 3, 0, 3, 2, 1, 3,
1, 2, 1, 2, 3, 2, 2, 1, 3, 3, 0, 0, 2, 1, 0, 0, 3, 0, 3, 2, 1, 1, 2,
2, 1, 2, 0, 3, 2, 0, 3, 1, 3, 0, 2, 0, 1, 1, 1, 1, 3, 3, 1, 2, 3, 0,
2, 3, 3, 2, 0, 0, 1, 2, 2, 3, 0, 1, 3, 2, 1, 2, 0, 0, 3, 3, 2, 0, 0,
0, 2, 1, 1, 0, 0, 2, 0, 0, 0, 1, 3, 1, 2, 0, 0, 1, 1, 1, 0, 0, 2, 1,
3], dtype=int32)
```

1.1.7 To predict the result on training set X

```
In [28]: y_kmeans = kmeans.predict(x)
```

1.1.8 To check the result on the trainning set X

```
In [24]: from sklearn import metrics
```

```
In [29]: print(metrics.confusion_matrix(y_true, y_kmeans))
```

```
[[ 0  0 75  0]
 [75  0  0  0]
 [ 0  0  1 74]
 [ 0 75  0  0]]
```

1.1.9 Plot the graph again with the plot cluster centers

```
In [38]: plt.scatter(x[:, 0], x[:,1], c='black', s=50, cmap='viridis')
```

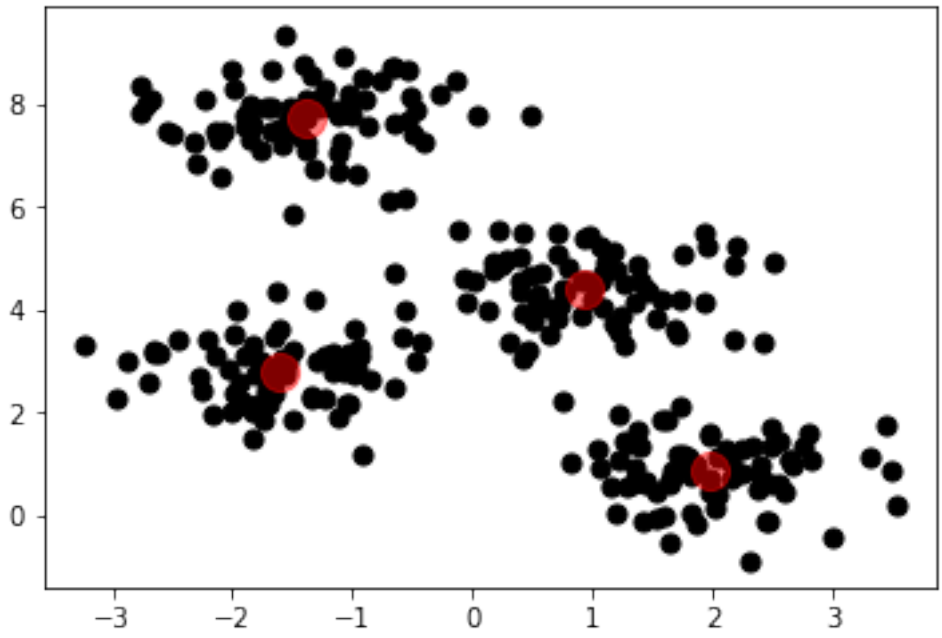
```
Out[38]: <matplotlib.collections.PathCollection at 0x7fd570cb2198>
```

```
In [39]: centers = kmeans.cluster_centers_
```

```
In [40]: plt.scatter(centers[:,0], centers[:,1], c='red', s=200, alpha=0.5)
```

```
Out[40]: <matplotlib.collections.PathCollection at 0x7fd570cb2c50>
```

```
In [41]: plt.show()
```


8
6
4
2
0
−3
−2
−1
0
1
2
3