Spherical K-Means

Name - Amith Korada

Course - AI & ML
(Batch - 4)

Duration - 12 Months

Problem Statement - Implement K-Means clustering and then use the clusters for classification purposes. Now using the same dataset, implement spherical clustering and then check accuracy for classification

Prerequisites -

What things you need to install the software and how to install them:

Python 3.6 This setup requires that your machine has the latest version of python. The following URL https://www.python.org/downloads/ can be referred to as download python.

The second and easier option is to download anaconda and use its anaconda prompt to run the commands. To install anaconda check this URL https://www.anaconda.com/download/ You will also need to download and install the below 3 packages after you install either python or anaconda from the steps above Sklearn (scikit-learn) numpy scipy if you have chosen to install python 3.6 then run the below commands in command prompt/terminal to install these packages pip install -U sci-kit-learn pip install NumPy pip install scipy if you have chosen to install anaconda then run the below commands in anaconda prompt to install these packages conda install -c sci-kit-learn conda install -c anaconda numpy conda install -c anaconda scipy.

1. Importing necessary libraries-

```
import numpy as np
from sklearn.datasets import make_classification
from sklearn.cluster import KMeans, SpectralClustering
from sklearn.metrics import accuracy_score
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
```

2. Generate a dummy dataset using Scikit-Learn (Features=10 and Classes =4)-

```
X, y = make_classification(n_samples=200, n_features=10,n_classes=4,n_informative=4)
```

```
X.shape
```

```
(200, 10)
```

```
y.shape
```

```
(200,)
```

```
np.unique(y, return_counts=True)
```

```
(array([0, 1, 2, 3]), array([49, 50, 51, 50], dtype=int64))
```

3. Implement K-Means clustering-

```
kmeans = KMeans(n_clusters=4)
```

```
pred = kmeans.fit(X)
```

```
predicted = pred.labels_
```

```
accuracy_score(y, predicted)
```

```
0.345
```

4. Implement spherical clustering-

```
SkMeans = SpectralClustering(n_clusters=4)
```

```
Spred = SkMeans.fit(X)
```

```
SPrediction = Spred.labels_
```

```
accuracy_score(y, SPrediction)
```

```
0.235
```

5. Projecting data into two dimensions using PCA-

```
pca = PCA(n_components=2)
```

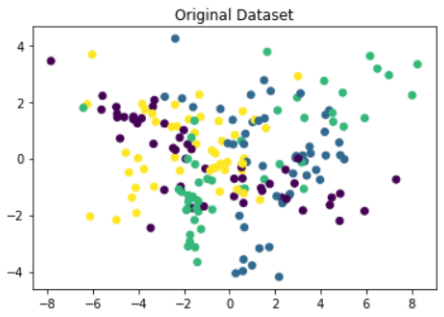
```
X_trans = pca.fit_transform(X)
```

```
X_trans.shape
```

```
(200, 2)
```

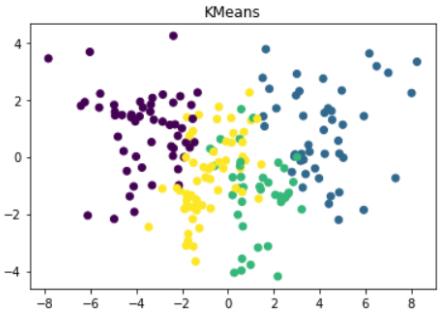
6. Visualising the original data-

```
plt.scatter(X_trans[:,0], X_trans[:,1], c=y)
plt.title("Original Dataset")
plt.show()
```



7. Clusters after KMeans-

```
plt.scatter(X_trans[:,0], X_trans[:,1], c=predicted)
plt.title("KMeans")
plt.show()
```



8. Clusters after Spherical KMeans-

```
plt.scatter(X_trans[:,0], X_trans[:,1], c=SPrediction)
plt.title("Spherical KMeans")
plt.show()
```