

# DIGITAL ASSIGNMENT

## KMEANS CLUSTERING

NAME SIDHANTHA PODDAR  
ROLLNO 17BCE2044

In [103]:

```
%matplotlib inline
from copy import deepcopy
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
```

In [2]:

```
data = pd.read_csv('East+Midlands.csv')
print(data.shape)
data.head()
```

(23677, 26)

Out[2]:

	Year	CP	Estimation_method	Estimation_method_detailed	Region	LocalAuthority	Roa
0	2000	6004	Counted	Manual count	East Midlands	Northamptonshire	M
1	2000	6005	Counted	Manual count	East Midlands	Leicestershire	M
2	2000	6006	Counted	Manual count	East Midlands	Derbyshire	M
3	2000	6043	Counted	Manual count	East Midlands	Northamptonshire	M4
4	2000	6082	Estimated	Estimated using previous year's AADF on this link	East Midlands	Rutland	A

5 rows × 26 columns

In [94]:

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs

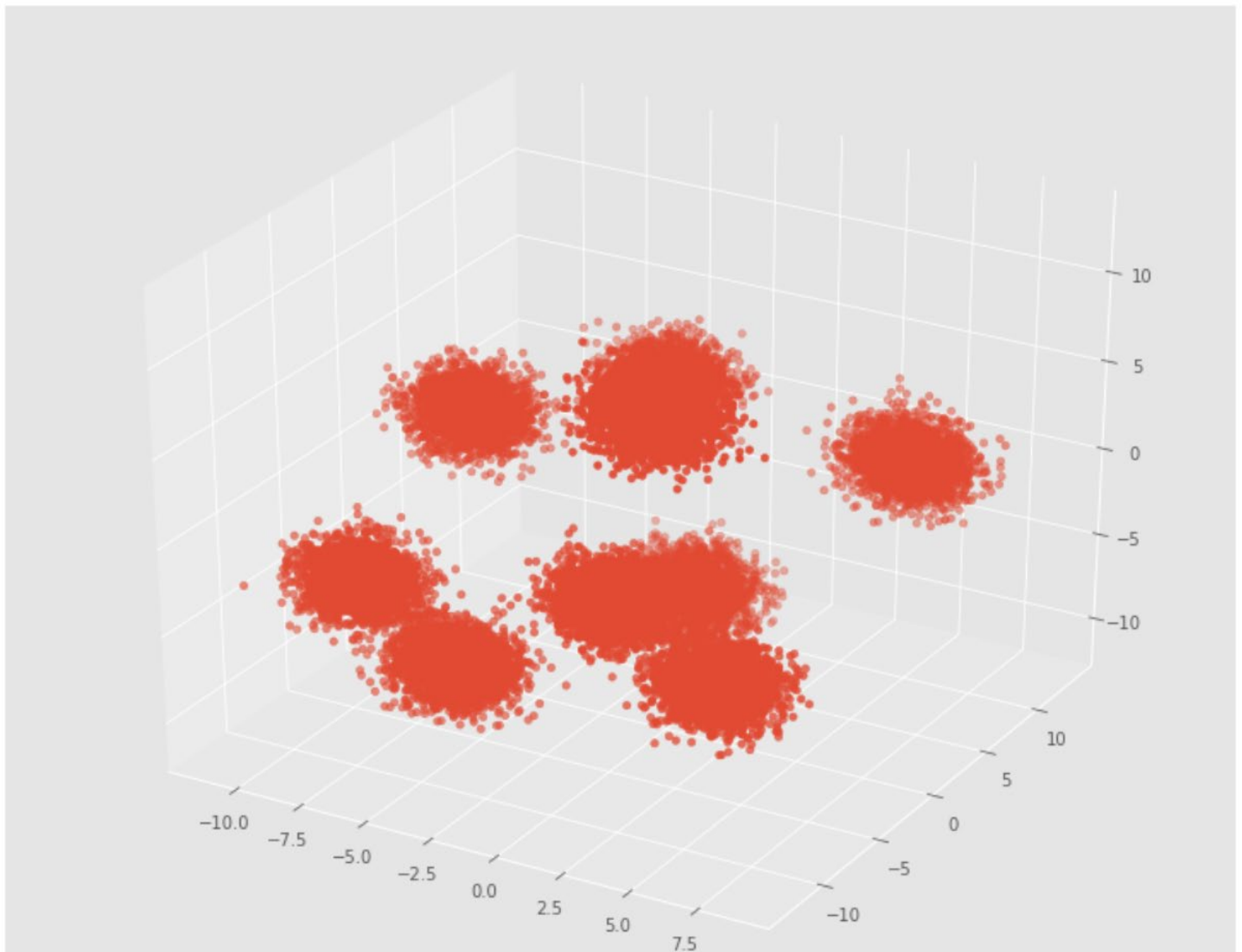
plt.rcParams['figure.figsize'] = (10,8)
X=data
# Creating a sample dataset with 4 clusters
X,Y = make_blobs(n_samples=23677, n_features=25, centers=9)
```

In [102]:

```
fig = plt.figure()
ax = Axes3D(fig)
ax.scatter(X[:, 0], X[:, 1], X[:, 2],X[:,3],s=20)
```

Out[102]:

<mpl\_toolkits.mplot3d.art3d.Path3DCollection at 0x1f33a7cb748>



In [94]:



In [96]:



```
# Initializing KMeans
kmeans = KMeans(n_clusters=5)
# Fitting with inputs
kmeans = kmeans.fit(X)
# Predicting the clusters
labels = kmeans.predict(X)
# Getting the cluster centers
C = kmeans.cluster_centers_
```

In [97]:



C

Out[97]:

```

array([[ -2.11662986e+00,  -8.68115769e+00,  -4.31173051e+00,
         6.67963995e+00,  -4.32489287e+00,   4.08539729e+00,
        -1.00496743e+00,  -4.04598438e+00,  -4.66583377e+00,
        -5.54089195e-01,   4.79366686e+00,   2.45084516e+00,
        -6.33107822e-01,   6.47012396e+00,  -2.85497840e+00,
         7.48780978e+00,  -2.89950329e+00,  -4.48614949e+00,
         1.60276744e+00,  -7.33902372e+00,   2.45827226e-01,
        -1.06700274e+00,  -6.11940909e-01,   2.46288221e+00,
        -4.09258300e+00],
 [ 1.52972804e-01,   7.76156845e+00,  -1.62240869e+00,
        -2.84969310e-01,  -2.16645399e+00,   3.53919329e-01,
         8.14015904e+00,  -3.52788920e-03,   8.21409183e-01,
         6.39890521e+00,  -5.28296073e+00,   1.11244317e+00,
         9.54035033e-01,   3.09461297e+00,   2.98482207e+00,
         3.90281709e-01,   4.55691726e+00,   4.75728727e+00,
         1.19142329e+00,  -3.71171613e+00,  -4.48806628e-01,
         7.42226211e+00,  -3.73692612e+00,  -2.14356143e+00,
         2.00060532e+00],
 [ 2.70007075e+00,  -7.34645066e+00,   9.47912842e+00,
        -4.99390956e+00,   8.19877051e+00,  -5.77024994e+00,
        -9.58381480e-01,  -2.09661172e+00,  -4.95137149e+00,
        -2.69801893e+00,  -7.71075382e+00,  -6.53697071e+00,
         9.31663497e+00,   5.78838382e+00,  -4.92993026e-01,
        -9.15415298e+00,   8.19200108e+00,  -9.78441045e+00,
        -9.01931179e+00,  -4.09833885e+00,   6.86111327e+00,
        -2.43148350e-01,   8.45586457e+00,   8.59390989e+00,
        -8.66331199e+00],
 [ -7.80733759e+00,   7.60495571e-01,   2.60277409e+00,
         1.47727916e+00,  -5.15093618e+00,  -8.48828605e+00,
         2.57012139e+00,  -6.11357692e+00,   3.38973182e+00,
         3.27485562e+00,  -9.88805928e+00,  -6.06451866e+00,
        -6.81710875e+00,   1.93520022e-01,  -5.93078179e+00,
         1.70225836e+00,   1.50804324e+00,  -8.99041567e+00,
         1.83400964e+00,  -7.33748593e+00,   7.26989946e-01,
        -9.98797615e+00,  -2.56075180e+00,  -9.38973522e+00,
         7.84706804e+00],
 [ 1.50674055e+00,  -8.04693763e+00,  -9.65127874e-01,
        -7.05675768e+00,   7.93162373e+00,   9.92436429e+00,
         5.31864292e+00,  -8.07273763e+00,  -4.32119380e+00,
        -8.28539390e+00,   7.59440341e+00,  -3.30105462e+00,
        -3.68258631e+00,   2.65738559e+00,  -4.50556156e+00,
         5.60250589e+00,  -8.20269113e+00,   9.46359654e+00,
        -4.47003939e+00,  -9.14791399e-01,   1.37763257e+00,
         5.74533954e+00,   6.05745607e+00,   2.53629442e+00,
         3.30053898e+00]])

```

In [98]:

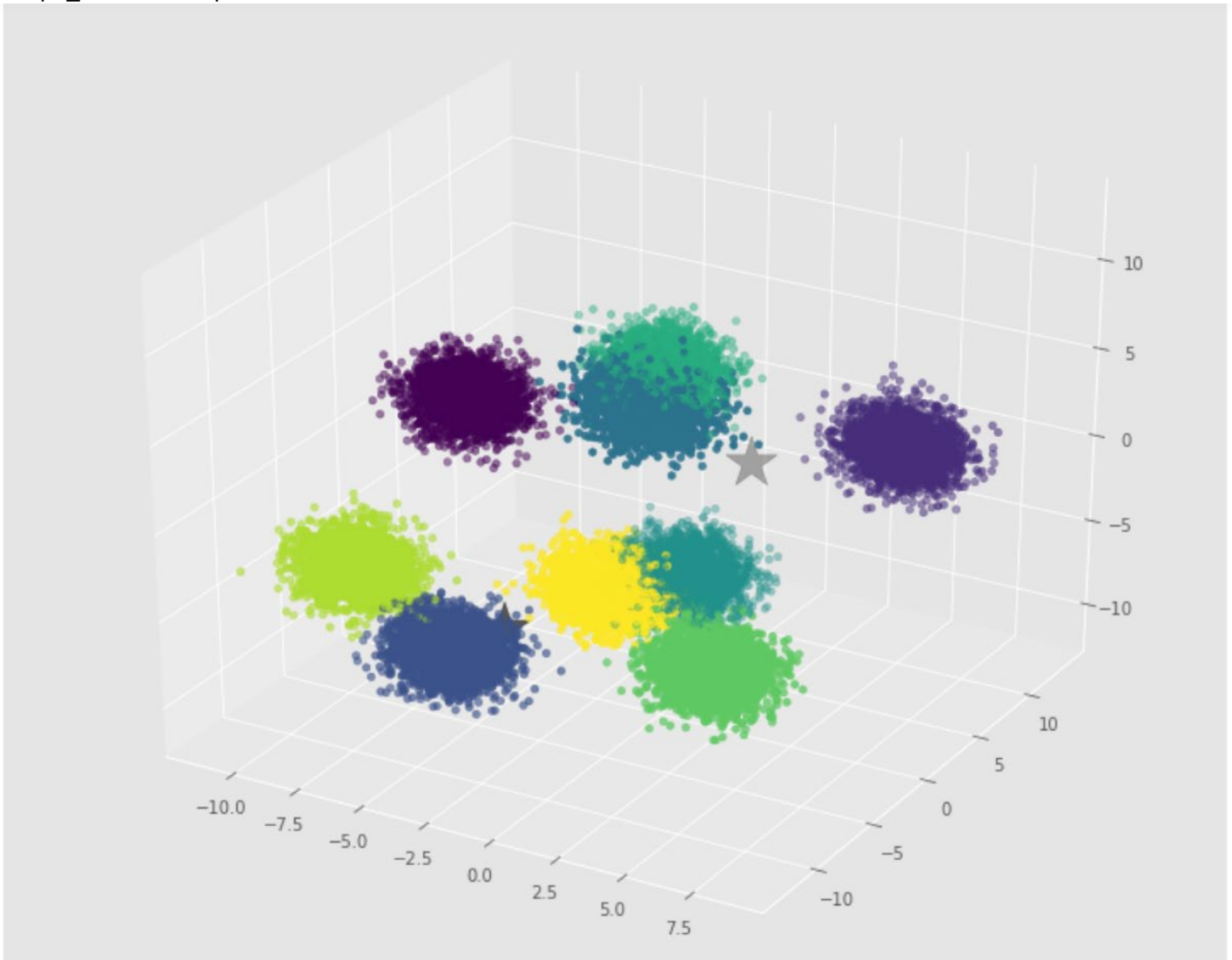


```
fig = plt.figure()
ax = Axes3D(fig)
ax.scatter(X[:, 0], X[:, 1], X[:, 2], X[:, 3], c=Y)
ax.scatter(C[:, 0], C[:, 1], C[:, 2], X[:, 3], marker='*', c='#050505', s=1000)
```

C:\Users\Sid\AppData\Local\conda\conda\envs\image\lib\site-packages\mpl\_toolkits\mplot3d\art3d.py:728: FutureWarning: elementwise comparison failed; returning scalar instead, but in the future will perform elementwise comparison  
if zdir == 'x':  
C:\Users\Sid\AppData\Local\conda\conda\envs\image\lib\site-packages\mpl\_toolkits\mplot3d\art3d.py:730: FutureWarning: elementwise comparison failed; returning scalar instead, but in the future will perform elementwise comparison  
elif zdir == 'y':

Out[98]:

&lt;mpl\_toolkits.mplot3d.art3d.Path3DCollection at 0x1f32c37da20&gt;



In [ ]:

