Experiment 8 Clustering Model

211090073 Astik Sonawane

In [27]:
```
import pandas as pd
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt
import scipy.cluster.hierarchy as sch
from sklearn.cluster import AgglomerativeClustering
from sklearn.cluster import KMeans
```

In [11]:
```
Univ = pd.read_csv(r'C:\Users\HOME\Desktop\DSA\Lab8\Universities.csv')
Univ.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Univ      25 non-null     object
 1   SAT       25 non-null     int64 
 2   Top10     25 non-null     int64 
 3   Accept    25 non-null     int64 
 4   SFRatio   25 non-null     int64 
 5   Expenses  25 non-null     int64 
 6   GradRate  25 non-null     int64 
dtypes: int64(6), object(1)
memory usage: 1.5+ KB
```

In [12]:
```
Univ.head()
```

Out[12]:

	Univ	SAT	Top10	Accept	SFRatio	Expenses	GradRate
0	Brown	1310	89	22	13	22704	94
1	CalTech	1415	100	25	6	63575	81
2	CMU	1260	62	59	9	25026	72
3	Columbia	1310	76	24	12	31510	88
4	Cornell	1280	83	33	13	21864	90

In [14]:
```
# Normalization function (custom Function)
def norm_func(i):
    x=(i-i.min())/(i.max()-i.min())
    return (x)
```

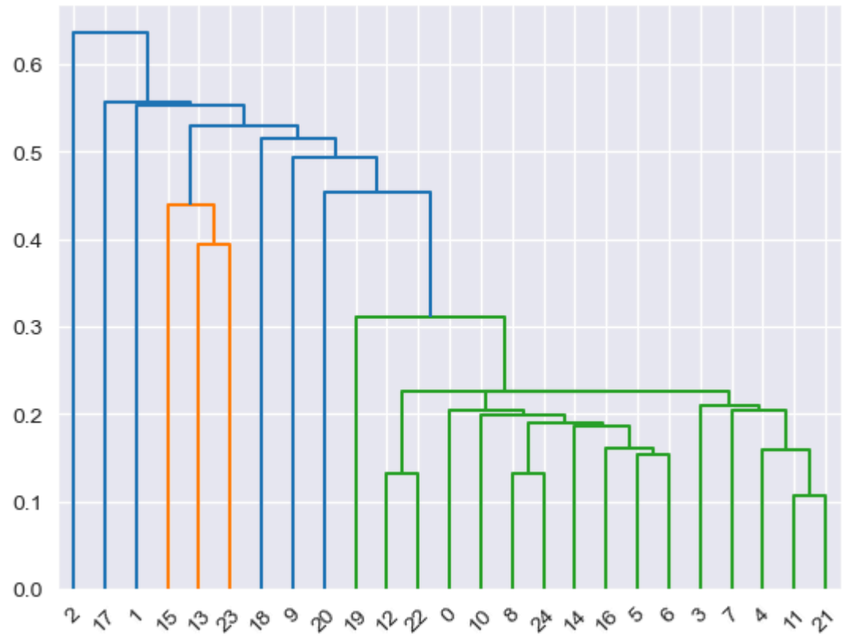
In [17]:
```
#normalized dataframe(considering numerical part of data)
df_norm = norm_func(Univ.iloc[:,1:])
```

In [18]:
```
df_norm
```

Out[18]:

	SAT	Top10	Accept	SFRatio	Expenses	GradRate
0	0.743902	0.847222	0.105263	0.368421	0.255144	0.900000
1	1.000000	1.000000	0.144737	0.000000	1.000000	0.466667
2	0.621951	0.472222	0.592105	0.157895	0.297461	0.166667
3	0.743902	0.666667	0.131579	0.315789	0.415629	0.700000
4	0.670732	0.763889	0.250000	0.368421	0.239835	0.766667
5	0.817073	0.847222	0.118421	0.210526	0.427512	0.933333
6	0.756098	0.861111	0.210526	0.315789	0.416996	0.933333
7	0.609756	0.638889	0.131579	0.315789	0.208161	0.833333
8	0.963415	0.875000	0.000000	0.263158	0.561699	1.000000
9	0.731707	0.652778	0.394737	0.052632	0.910991	0.666667
10	0.914634	0.916667	0.210526	0.210526	0.476864	0.800000
11	0.621951	0.791667	0.328947	0.263158	0.352609	0.733333
12	0.609756	0.736111	0.368421	0.368421	0.116965	0.900000
13	0.185366	0.138889	0.526316	0.631579	0.026991	0.433333
14	0.902439	0.875000	0.000000	0.105263	0.392120	0.933333
15	0.000000	0.000000	1.000000	0.684211	0.006597	0.066667
16	0.865854	0.861111	0.078947	0.315789	0.505659	0.866667
17	0.170732	0.291667	0.697368	1.000000	0.000000	0.000000
18	0.573171	0.930556	0.342105	0.578947	0.117293	0.366667
19	0.695122	0.652778	0.473684	0.368421	0.540832	0.666667
20	0.426829	0.513889	0.710526	0.526316	0.123307	0.600000
21	0.682927	0.722222	0.289474	0.263158	0.343515	0.766667
22	0.536585	0.680556	0.394737	0.421053	0.084653	0.833333
23	0.195122	0.166667	0.723684	0.473684	0.057462	0.133333
24	0.902439	0.930556	0.065789	0.263158	0.634397	0.966667

In [19]:
```
#create dendrogram
dendrogram = sch.dendrogram(sch.linkage(df_norm,method='single'))
```



In [20]:
```
#create clusters
hc = AgglomerativeClustering(n_clusters=4,affinity='euclidean',linkage='single')
```

In [21]:
```
# save clusters for chart
y_hc = hc.fit_predict(df_norm)
Clusters = pd.DataFrame(y_hc,columns=['Clusters'])
```

```
c:\Users\HOME\anaconda3\Lib\site-packages\sklearn\cluster\_agglomerative.py:1005: FutureWarning: Attribute `affinity` was deprecated in version 1.2 and will be removed in 1.4. Use `me
tric` instead
  warnings.warn(
```

In [22]:
```
Clusters
```

Out[22]:

	Clusters
0	0
1	3
2	1
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	2
18	0
19	0
20	0
21	0
22	0
23	0
24	0

In [26]:
```
df_norm['h_clusterid']=Clusters
Univ['h_clusterid']=Clusters
Univ.head()
```

Out[26]:

	Univ	SAT	Top10	Accept	SFRatio	Expenses	GradRate	h_clusterid
0	Brown	1310	89	22	13	22704	94	0
1	CalTech	1415	100	25	6	63575	81	3
2	CMU	1260	62	59	9	25026	72	1
3	Columbia	1310	76	24	12	31510	88	0
4	Cornell	1280	83	33	13	21864	90	0

K-Means

In [51]:
```
Univ = pd.read_csv(r'C:\Users\HOME\Desktop\DSA\Lab8\Universities.csv')
Univ
```

Out[51]:

	Univ	SAT	Top10	Accept	SFRatio	Expenses	GradRate
0	Brown	1310	89	22	13	22704	94
1	CalTech	1415	100	25	6	63575	81
2	CMU	1260	62	59	9	25026	72
3	Columbia	1310	76	24	12	31510	88
4	Cornell	1280	83	33	13	21864	90
5	Dartmouth	1340	89	23	10	32162	95
6	Duke	1315	90	30	12	31585	95
7	Georgetown	1255	74	24	12	20126	92
8	Harvard	1400	91	14	11	39525	97
9	JohnsHopkins	1305	75	44	7	58691	87
10	MIT	1380	94	30	10	34870	91
11	Northwestern	1260	85	39	11	28052	89
12	NotreDame	1255	81	42	13	15122	94
13	PennState	1081	38	54	18	10185	80
14	Princeton	1375	91	14	8	30220	95
15	Purdue	1005	28	90	19	9066	69
16	Stanford	1360	90	20	12	36450	93
17	TexasA&M	1075	49	67	25	8704	67
18	UCBerkeley	1240	95	40	17	15140	78
19	UChicago	1290	75	50	13	38380	87
20	UMichigan	1180	65	68	16	15470	85
21	UPenn	1285	80	36	11	27553	90
22	UVA	1225	77	44	14	13349	92
23	UWisconsin	1085	40	69	15	11857	71
24	Yale	1375	95	19	11	43514	96

In [52]:
```
#normalization function
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaled_Univ_df = scaler.fit_transform(Univ.iloc[:,1:])
```

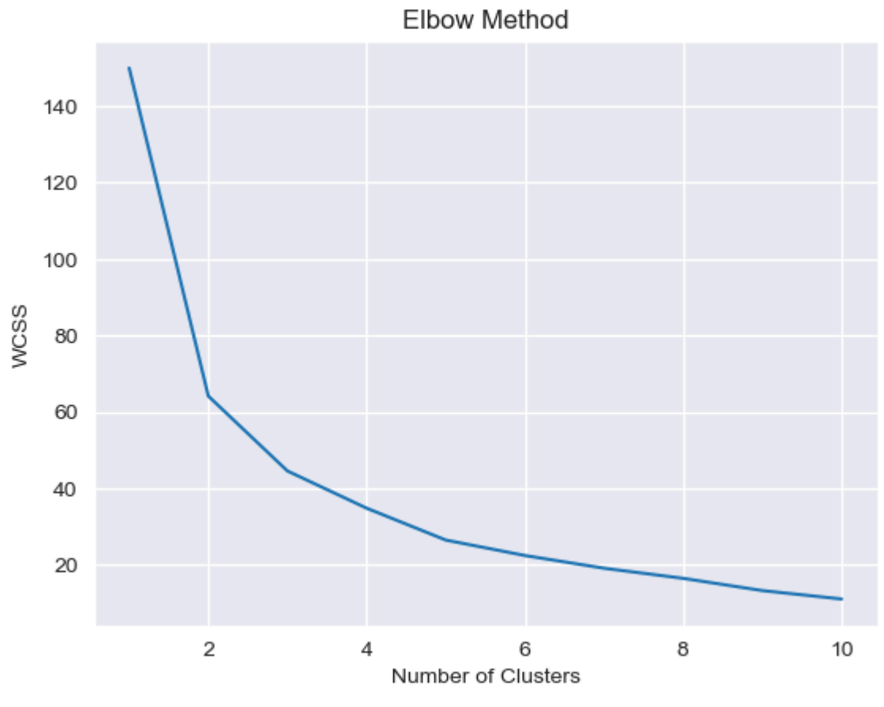
In [53]:
```
#to find optimum number of cluster
# Kmeans aim to choose centroids that minimise inertia or with cluster sum of squares criterion
```

In [54]:
```
wcss = []
for i in range(1,11):
    kmeans = KMeans(n_clusters=i,random_state=0)
    kmeans.fit(scaled_Univ_df)
    wcss.append(kmeans.inertia_)

plt.plot(range(1,11),wcss) #wcss=cluster sum of squeares
plt.title('Elbow Method')
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.show()
```

```
c:\Users\HOME\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
c:\Users\HOME\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than avai
lable threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\HOME\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
c:\Users\HOME\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than avai
lable threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\HOME\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
c:\Users\HOME\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than avai
lable threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\HOME\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
c:\Users\HOME\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than avai
lable threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\HOME\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
c:\Users\HOME\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than avai
lable threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\HOME\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
c:\Users\HOME\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than avai
lable threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\HOME\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
c:\Users\HOME\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than avai
lable threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\HOME\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
c:\Users\HOME\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than avai
lable threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
c:\Users\HOME\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```

```
c:\Users\HOME\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than avai
lable threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```



In [55]:
```
#build cluster algorithm
from sklearn.cluster import KMeans
clusters_new = KMeans(4,random_state=42)
clusters_new.fit(scaled_Univ_df)
```

```
c:\Users\HOME\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
c:\Users\HOME\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than avai
lable threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

Out[55]:
```
KMeans(n_clusters=4, random_state=42)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [56]:
```
clusters_new.labels_
```

Out[56]:
```
array([1, 3, 2, 1, 2, 1, 1, 2, 1, 3, 1, 2, 2, 0, 1, 0, 1, 0, 2, 2, 2, 2,
       2, 0, 1])
```

In [57]:
```
#assign clusters to data set
Univ['clusterid_new'] = clusters_new.labels_
```

In [58]:
```
# Standardized values
clusters_new.cluster_centers_
```

Out[58]:
```
array([[-1.93029211, -1.98148647,  1.59348244,  1.63857398, -1.23359906,
        -1.68680366],
       [ 0.80273428,  0.68086062, -0.90136381, -0.43159988,  0.44062556,
         0.79526289],
       [-0.12658888,  0.06407139,  0.2224667 ,  0.04516743, -0.38064332,
         0.02028221],
       [ 0.88122441,  0.5787432 , -0.24316128, -1.56078563,  2.38759968,
        -0.3064867 ]])
```

In [59]:
```
# Assuming you want to drop the column named 'clusterid_new'
Univ.drop(columns=['Univ'], inplace=True)
```

In [60]:
```
Univ.groupby('clusterid_new').agg(['mean']).reset_index()
```

Out[60]:

clusterid_new		SAT	Top10	Accept	SFRatio	Expenses	GradRate
		mean	mean	mean	mean	mean	mean
0	0	1061.500000	38.750000	70.000000	19.25	9953.000000	71.750000
1	1	1351.666667	89.444444	21.777778	11.00	33615.555556	93.777778
2	2	1253.000000	77.700000	43.500000	12.90	22008.200000	86.900000
3	3	1360.000000	87.500000	34.500000	6.50	61133.000000	84.000000

In [61]:
```
Univ.head()
```

Out[61]:

	SAT	Top10	Accept	SFRatio	Expenses	GradRate	clusterid_new
0	1310	89	22	13	22704	94	1
1	1415	100	25	6	63575	81	3
2	1260	62	59	9	25026	72	2
3	1310	76	24	12	31510	88	1
4	1280	83	33	13	21864	90	2

In [ ]: