Problem Statement:-

Perform clustering (hierarchical,K means clustering and DBSCAN) for the airlines data to obtain optimum number of clusters, Draw the inferences from the clusters obtained.

Hierarchical Clustering

1. Import Neccesary Libraries

In [1]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
```

2. Import Data

In [2]:
```
Airline_Data = pd.read_csv('EastWestAirlines.csv')
Airline_Data
```

Out[2]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
0	1	28143	0	1	1	1	174	1
1	2	19244	0	1	1	1	215	2
2	3	41354	0	1	1	1	4123	4
3	4	14776	0	1	1	1	500	1
4	5	97752	0	4	1	1	43300	26
...	...	...	...	...	...	...	...	...
3994	4017	18476	0	1	1	1	8525	4
3995	4018	64385	0	1	1	1	981	5
3996	4019	73597	0	3	1	1	25447	8
3997	4020	54899	0	1	1	1	500	1
3998	4021	3016	0	1	1	1	0	0

3. Data Understanding

In [3]:
```
Airline_Data.shape
```

Out[3]: (3999, 12)

In [4]:
```
Airline_Data.isna().sum()
```

Out[4]:
```
ID#                  0
Balance              0
Qual_miles           0
cc1_miles            0
cc2_miles            0
cc3_miles            0
Bonus_miles          0
Bonus_trans          0
Flight_miles_12mo    0
Flight_trans_12      0
Days_since_enroll    0
Award?               0
dtype: int64
```

In [5]:
```
Airline_Data.dtypes
```

Out[5]:
```
ID#                  int64
Balance              int64
Qual_miles           int64
cc1_miles            int64
cc2_miles            int64
cc3_miles            int64
Bonus_miles          int64
Bonus_trans          int64
Flight_miles_12mo    int64
Flight_trans_12      int64
Days_since_enroll    int64
Award?               int64
dtype: object
```

In [6]:
```
Airline_Data.describe()
```

Out[6]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_n
count	3999.000000	3.999000e+03	3999.000000	3999.000000	3999.000000	3999.000000	3999.00(
mean	2014.819455	7.360133e+04	144.114529	2.059515	1.014504	1.012253	17144.84(
std	1160.764358	1.007757e+05	773.663804	1.376919	0.147650	0.195241	24150.96
min	1.000000	0.000000e+00	0.000000	1.000000	1.000000	1.000000	0.00(
25%	1010.500000	1.852750e+04	0.000000	1.000000	1.000000	1.000000	1250.00(
50%	2016.000000	4.309700e+04	0.000000	1.000000	1.000000	1.000000	7171.00(
75%	3020.500000	9.240400e+04	0.000000	3.000000	1.000000	1.000000	23800.50(
max	4021.000000	1.704838e+06	11148.000000	5.000000	3.000000	5.000000	263685.00(

4. Data Preparation

Data Normalization

In [7]:
```
from sklearn.preprocessing import normalize
Airline_Data_N = pd.DataFrame(normalize(Airline_Data), columns=Airline_Data.colum
Airline_Data_N
```

Out[7]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
0	0.000034	0.970414	0.0	0.000034	0.000034	0.000034	0.006000	0.000034
1	0.000098	0.940209	0.0	0.000049	0.000049	0.000049	0.010504	0.000098
2	0.000071	0.981113	0.0	0.000024	0.000024	0.000024	0.097817	0.000095
3	0.000245	0.904428	0.0	0.000061	0.000061	0.000061	0.030605	0.000061
4	0.000047	0.912226	0.0	0.000037	0.000009	0.000009	0.404078	0.000243
...	...	...	...	...	...	...	...	...
3994	0.193227	0.888739	0.0	0.000048	0.000048	0.000048	0.410072	0.000192
3995	0.062263	0.997710	0.0	0.000015	0.000015	0.000015	0.015202	0.000077
3996	0.051533	0.943692	0.0	0.000038	0.000013	0.000013	0.326292	0.000103
3997	0.073000	0.996925	0.0	0.000018	0.000018	0.000018	0.009080	0.000018
3998	0.770721	0.578089	0.0	0.000192	0.000192	0.000192	0.000000	0.000000

3999 rows × 12 columns

In [28]:
```
Airline_Data_N.mean()
```

Out[28]:
```
ID#                  0.092613
Balance              0.881585
Qual_miles           0.002808
cc1_miles            0.000053
cc2_miles            0.000039
cc3_miles            0.000039
Bonus_miles          0.232972
Bonus_trans          0.000263
Flight_miles_12mo    0.007778
Flight_trans_12      0.000025
Days_since_enroll    0.129491
Award?               0.000009
Clusters             0.002001
dtype: float64
```

In [29]:
```
Airline_Data_N.std()
```

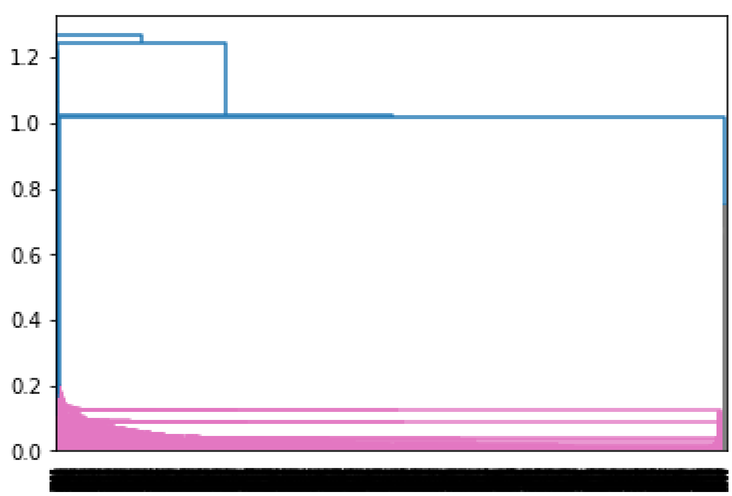
Out[29]:
```
ID#                  0.152652
Balance              0.185919
Qual_miles           0.021861
cc1_miles            0.000047
cc2_miles            0.000047
cc3_miles            0.000047
Bonus_miles          0.242524
Bonus_trans          0.000337
Flight_miles_12mo    0.027704
Flight_trans_12      0.000106
Days_since_enroll    0.158812
Award?               0.000022
Clusters             0.067069
dtype: float64
```

Dendrogram with all the linkages

In [116]:
```
import scipy.cluster.hierarchy as sch
from sklearn.cluster import AgglomerativeClustering
```

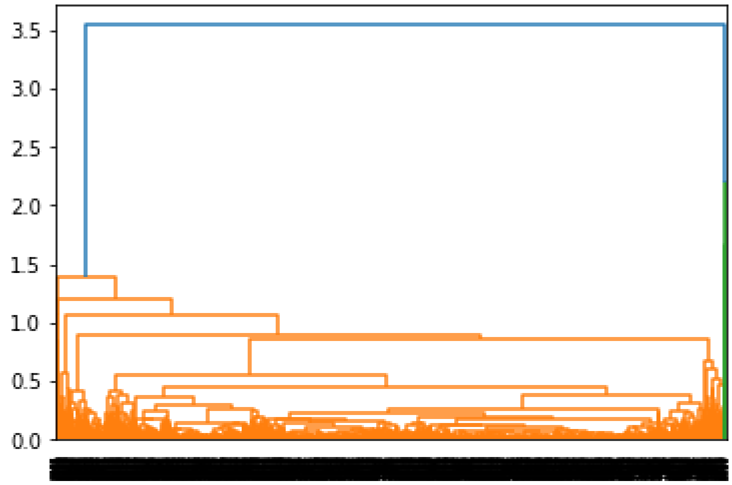
1. For Single linkage

In [117]:
```
dendrogram = sch.dendrogram(sch.linkage(Airline_Data_N, method='single'))
```



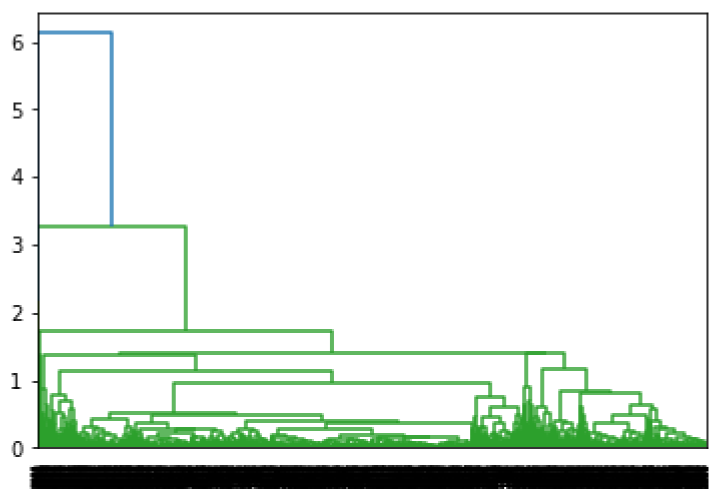
2. For Average linkage

In [118]:
```
dendrogram = sch.dendrogram(sch.linkage(Airline_Data_N, method='average'))
```



3. For Complete linkage

In [119]:
```
dendrogram = sch.dendrogram(sch.linkage(Airline_Data_N, method='complete'))
```



3. For Complete linkage

5. Model Building

In [41]:
```
HC = AgglomerativeClustering(n_clusters=5, affinity='euclidean', linkage='single'
HC
```

Out[41]:
```
AgglomerativeClustering(linkage='single', n_clusters=5)
```

6. Model Training

In [42]:
```
AgglomerativeClustering=HC.fit_predict(Airline_Data_N)
AgglomerativeClustering
```

Out[42]:
```
array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

In [43]:
```
pd.DataFrame(AgglomerativeClustering)
```

Out[43]:

	0
0	0
1	0
2	0
3	0
4	0
...	...
3994	0
3995	0
3996	0
3997	0
3998	0

3999 rows × 1 columns

In [44]:
```
HC_Clusters_Data = pd.DataFrame(AgglomerativeClustering, columns=['Clusters'])
HC_Clusters_Data
```

Out[44]:

	Clusters
0	0
1	0
2	0
3	0
4	0
...	...
3994	0
3995	0
3996	0
3997	0
3998	0

3999 rows × 1 columns

In [45]:
```
Airline_Data_N['Clusters']=HC_Clusters_Data['Clusters']
Airline_Data_N
```

Out[45]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
0	0.000034	0.970414	0.0	0.000034	0.000034	0.000034	0.006000	0.000034
1	0.000098	0.940209	0.0	0.000049	0.000049	0.000049	0.010504	0.000098
2	0.000071	0.981113	0.0	0.000024	0.000024	0.000024	0.097817	0.000095
3	0.000245	0.904428	0.0	0.000061	0.000061	0.000061	0.030605	0.000061
4	0.000047	0.912226	0.0	0.000037	0.000009	0.000009	0.404078	0.000243
...	...	...	...	...	...	...	...	...
3994	0.193227	0.888739	0.0	0.000048	0.000048	0.000048	0.410072	0.000192
3995	0.062263	0.997710	0.0	0.000015	0.000015	0.000015	0.015202	0.000077
3996	0.051533	0.943692	0.0	0.000038	0.000013	0.000013	0.326292	0.000103
3997	0.073000	0.996925	0.0	0.000018	0.000018	0.000018	0.009080	0.000018
3998	0.770721	0.578089	0.0	0.000192	0.000192	0.000192	0.000000	0.000000

3999 rows × 13 columns

1. Hierarchical Clustering With Cluster '0'

In [46]:
```
Airline_Data_N[Airline_Data_N['Clusters']==0]
```

Out[46]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
0	0.000034	0.970414	0.0	0.000034	0.000034	0.000034	0.006000	0.000034
1	0.000098	0.940209	0.0	0.000049	0.000049	0.000049	0.010504	0.000098
2	0.000071	0.981113	0.0	0.000024	0.000024	0.000024	0.097817	0.000095
3	0.000245	0.904428	0.0	0.000061	0.000061	0.000061	0.030605	0.000061
4	0.000047	0.912226	0.0	0.000037	0.000009	0.000009	0.404078	0.000243
...	...	...	...	...	...	...	...	...
3994	0.193227	0.888739	0.0	0.000048	0.000048	0.000048	0.410072	0.000192
3995	0.062263	0.997710	0.0	0.000015	0.000015	0.000015	0.015202	0.000077
3996	0.051533	0.943692	0.0	0.000038	0.000013	0.000013	0.326292	0.000103
3997	0.073000	0.996925	0.0	0.000018	0.000018	0.000018	0.009080	0.000018
3998	0.770721	0.578089	0.0	0.000192	0.000192	0.000192	0.000000	0.000000

3994 rows × 13 columns

2. Hierarchical Clustering With Cluster '1'

In [52]:
```
Airline_Data_N[Airline_Data_N['Clusters']==1]
```

Out[52]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
2903	0.480596	0.476488	0.0	0.000164	0.000164	0.000164	0.476488	0.004765
3824	0.369484	0.528246	0.0	0.000096	0.000096	0.000096	0.528246	0.001056

3. Hierarchical Clustering With Cluster '2'

In [53]:
```
Airline_Data_N[Airline_Data_N['Clusters']==2]
```

Out[53]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
2793	0.445766	0.73682	0.385274	0.000158	0.000158	0.000158	0.0	0.0

4. Hierarchical Clustering With Cluster '3'

In [54]:
```
Airline_Data_N[Airline_Data_N['Clusters']==3]
```

Out[54]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
958	0.040749	0.562459	0.18087	0.000168	0.000042	0.000042	0.661578	0.001808

5. Hierarchical Clustering With Cluster '4'

In [55]:
```
Airline_Data_N[Airline_Data_N['Clusters']==4]
```

Out[55]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
2996	0.410687	0.0	0.0	0.000136	0.000136	0.000136	0.747211	0.002449

6. Hierarchical Clustering With Cluster '5'

In [56]:
```
Airline_Data_N[Airline_Data_N['Clusters']==5]
```

Out[56]:

ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_m

Hierarchical Clustering END

KMeans Clustering

In [60]:
```
Airline_Data_N
```

Out[60]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
0	0.000034	0.970414	0.0	0.000034	0.000034	0.000034	0.006000	0.000034
1	0.000098	0.940209	0.0	0.000049	0.000049	0.000049	0.010504	0.000098
2	0.000071	0.981113	0.0	0.000024	0.000024	0.000024	0.097817	0.000095
3	0.000245	0.904428	0.0	0.000061	0.000061	0.000061	0.030605	0.000061
4	0.000047	0.912226	0.0	0.000037	0.000009	0.000009	0.404078	0.000243
...	...	...	...	...	...	...	...	...
3994	0.193227	0.888739	0.0	0.000048	0.000048	0.000048	0.410072	0.000192
3995	0.062263	0.997710	0.0	0.000015	0.000015	0.000015	0.015202	0.000077
3996	0.051533	0.943692	0.0	0.000038	0.000013	0.000013	0.326292	0.000103
3997	0.073000	0.996925	0.0	0.000018	0.000018	0.000018	0.009080	0.000018
3998	0.770721	0.578089	0.0	0.000192	0.000192	0.000192	0.000000	0.000000

3999 rows × 13 columns

In [61]:
```
del Airline_Data_N['Clusters']
```

In [62]:
```
Airline_Data_N
```

Out[62]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
0	0.000034	0.970414	0.0	0.000034	0.000034	0.000034	0.006000	0.000034
1	0.000098	0.940209	0.0	0.000049	0.000049	0.000049	0.010504	0.000098
2	0.000071	0.981113	0.0	0.000024	0.000024	0.000024	0.097817	0.000095
3	0.000245	0.904428	0.0	0.000061	0.000061	0.000061	0.030605	0.000061
4	0.000047	0.912226	0.0	0.000037	0.000009	0.000009	0.404078	0.000243
...	...	...	...	...	...	...	...	...
3994	0.193227	0.888739	0.0	0.000048	0.000048	0.000048	0.410072	0.000192
3995	0.062263	0.997710	0.0	0.000015	0.000015	0.000015	0.015202	0.000077
3996	0.051533	0.943692	0.0	0.000038	0.000013	0.000013	0.326292	0.000103
3997	0.073000	0.996925	0.0	0.000018	0.000018	0.000018	0.009080	0.000018
3998	0.770721	0.578089	0.0	0.000192	0.000192	0.000192	0.000000	0.000000

3999 rows × 12 columns

In [67]:
```
from sklearn.cluster import KMeans
```

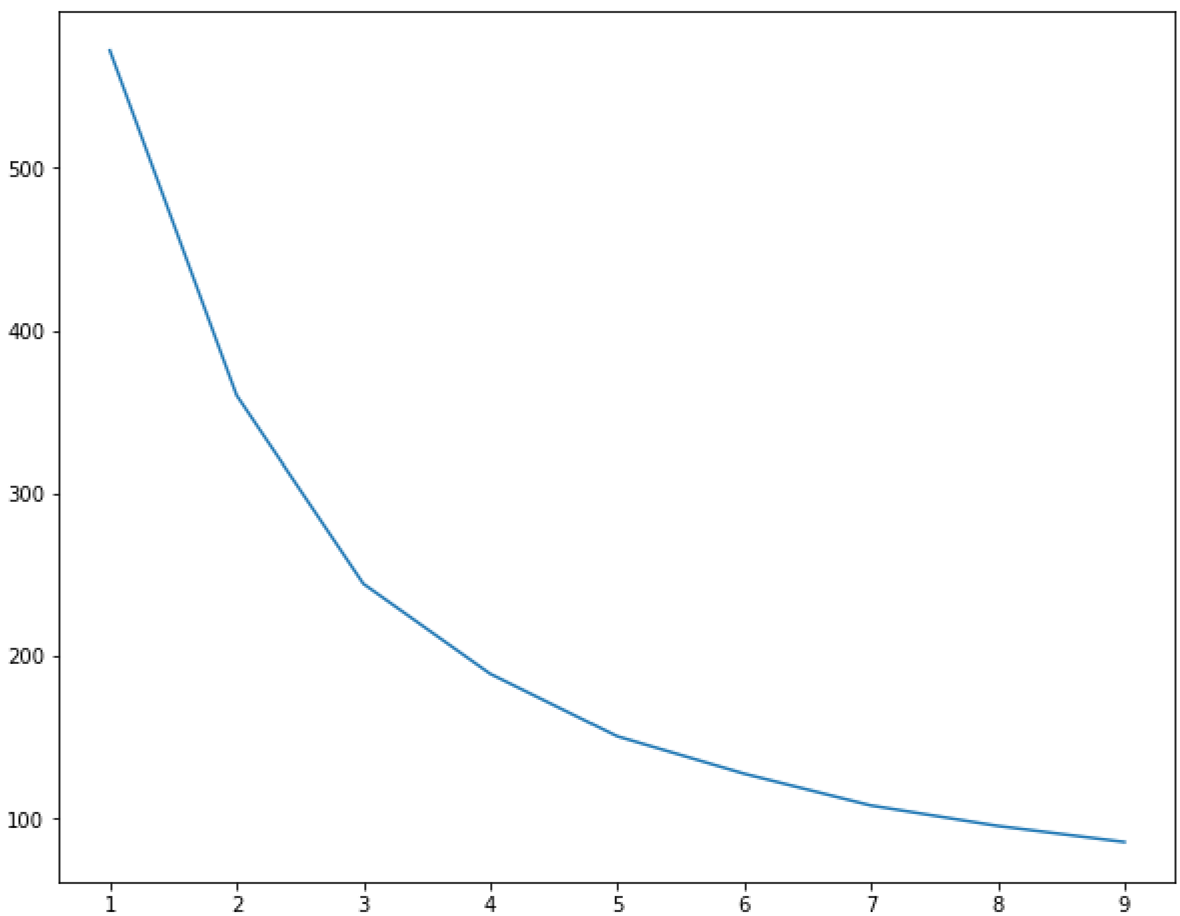
In [68]:
```
wcss = []
for i in range(1,10):
    Kmeans = KMeans(n_clusters=i)
    Kmeans.fit(Airline_Data_N)
    wcss.append(Kmeans.inertia_)
```

In [70]:
```
wcss
```

Out[70]:
```
[572.3268470917421,
 360.22431395114637,
 244.51900818247228,
 189.1474791179571,
 150.6886652638977,
 127.67559251102972,
 108.11456683323155,
 95.54429801795018,
 85.63458010208448]
```

Finding optimized value of n using Elbow method

In [74]:
```
plt.figure(figsize=(10,8))
plt.plot(range(1,10), wcss)
plt.show()
```



In [76]:
```
x = KMeans(n_clusters=4, max_iter=500, algorithm='auto')
x.fit(Airline_Data_N)
```

Out[76]:
```
KMeans(max_iter=500, n_clusters=4)
```

In [77]:
```
KMeans_Clusters = x.fit_predict(Airline_Data_N)
KMeans_Clusters
```

Out[77]:
```
array([2, 2, 2, ..., 1, 2, 0])
```

In [81]:
```
Airline_Data_N['K_Clusters']=KMeans_Clusters
Airline_Data_N
```

Out[81]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
0	0.000034	0.970414	0.0	0.000034	0.000034	0.000034	0.006000	0.000034
1	0.000098	0.940209	0.0	0.000049	0.000049	0.000049	0.010504	0.000098
2	0.000071	0.981113	0.0	0.000024	0.000024	0.000024	0.097817	0.000095
3	0.000245	0.904428	0.0	0.000061	0.000061	0.000061	0.030605	0.000061
4	0.000047	0.912226	0.0	0.000037	0.000009	0.000009	0.404078	0.000243
...	...	...	...	...	...	...	...	...
3994	0.193227	0.888739	0.0	0.000048	0.000048	0.000048	0.410072	0.000192
3995	0.062263	0.997710	0.0	0.000015	0.000015	0.000015	0.015202	0.000077
3996	0.051533	0.943692	0.0	0.000038	0.000013	0.000013	0.326292	0.000103
3997	0.073000	0.996925	0.0	0.000018	0.000018	0.000018	0.009080	0.000018
3998	0.770721	0.578089	0.0	0.000192	0.000192	0.000192	0.000000	0.000000

3999 rows × 13 columns

1. KMeans Clustering With Clusters '0'

In [82]:
```
Airline_Data_N[Airline_Data_N['K_Clusters']==0]
```

Out[82]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
27	0.002362	0.744599	0.0	0.000084	0.000084	0.000084	0.000000	0.000000
31	0.002510	0.786112	0.0	0.000078	0.000078	0.000078	0.000000	0.000000
39	0.004912	0.267216	0.0	0.000123	0.000123	0.000123	0.000000	0.000000
51	0.007429	0.185716	0.0	0.000143	0.000143	0.000143	0.052858	0.000143
66	0.005713	0.799352	0.0	0.000085	0.000085	0.000085	0.149212	0.000597
...	...	...	...	...	...	...	...	...
3984	0.930791	0.093846	0.0	0.000232	0.000232	0.000232	0.127760	0.000697
3988	0.581381	0.724733	0.0	0.000145	0.000145	0.000145	0.308011	0.000435
3989	0.763908	0.499244	0.0	0.000190	0.000190	0.000190	0.309409	0.001142
3993	0.688680	0.681477	0.0	0.000171	0.000171	0.000171	0.062592	0.000514
3998	0.770721	0.578089	0.0	0.000192	0.000192	0.000192	0.000000	0.000000

313 rows × 13 columns

2. KMeans Clustering With Clusters '1'

In [83]:
```
Airline_Data_N[Airline_Data_N['K_Clusters']==1]
```

Out[83]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
4	0.000047	0.912226	0.0	0.000037	0.000009	0.000009	0.404078	0.000243
6	0.000078	0.948504	0.0	0.000034	0.000011	0.000011	0.306979	0.000279
9	0.000092	0.963153	0.0	0.000028	0.000009	0.000009	0.261097	0.000257
11	0.000105	0.843373	0.0	0.000044	0.000009	0.000009	0.533912	0.000166
17	0.001114	0.864115	0.0	0.000062	0.000062	0.000062	0.265648	0.000310
...	...	...	...	...	...	...	...	...
3986	0.101902	0.870199	0.0	0.000025	0.000025	0.000025	0.480662	0.000178
3990	0.306688	0.864352	0.0	0.000076	0.000076	0.000076	0.383723	0.000153
3991	0.095285	0.929159	0.0	0.000071	0.000024	0.000024	0.355621	0.000665
3994	0.193227	0.888739	0.0	0.000048	0.000048	0.000048	0.410072	0.000192
3996	0.051533	0.943692	0.0	0.000038	0.000013	0.000013	0.326292	0.000103

1035 rows × 13 columns

3. KMeans Clustering With Clusters '2'

In [84]:
```
Airline_Data_N[Airline_Data_N['K_Clusters']==2]
```

Out[84]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
0	0.000034	0.970414	0.0	0.000034	0.000034	0.000034	0.006000	0.000034
1	0.000098	0.940209	0.0	0.000049	0.000049	0.000049	0.010504	0.000098
2	0.000071	0.981113	0.0	0.000024	0.000024	0.000024	0.097817	0.000095
3	0.000245	0.904428	0.0	0.000061	0.000061	0.000061	0.030605	0.000061
5	0.000337	0.921066	0.0	0.000056	0.000056	0.000056	0.000000	0.000000
...	...	...	...	...	...	...	...	...
3983	0.151070	0.987009	0.0	0.000038	0.000038	0.000038	0.011502	0.000038
3987	0.316486	0.941803	0.0	0.000079	0.000079	0.000079	0.019652	0.000237
3992	0.334609	0.931822	0.0	0.000083	0.000083	0.000083	0.077423	0.001000
3995	0.062263	0.997710	0.0	0.000015	0.000015	0.000015	0.015202	0.000077
3997	0.073000	0.996925	0.0	0.000018	0.000018	0.000018	0.009080	0.000018

2307 rows × 13 columns

4. KMeans Clustering With Clusters '3'

In [85]:
```
Airline_Data_N[Airline_Data_N['K_Clusters']==3]
```

Out[85]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
15	0.000278	0.495715	0.0	0.000070	0.000017	0.000017	0.860121	0.000261
16	0.000237	0.723935	0.0	0.000056	0.000014	0.000014	0.683099	0.000223
41	0.001043	0.259949	0.0	0.000099	0.000025	0.000025	0.945799	0.000646
58	0.001145	0.739082	0.0	0.000058	0.000019	0.000019	0.660413	0.000155
61	0.002258	0.725550	0.0	0.000036	0.000036	0.000036	0.641149	0.000401
...	...	...	...	...	...	...	...	...
3913	0.193810	0.638451	0.0	0.000148	0.000049	0.000049	0.739640	0.000788
3919	0.478383	0.606778	0.0	0.000121	0.000121	0.000121	0.606778	0.000121
3924	0.188174	0.704401	0.0	0.000048	0.000048	0.000048	0.680563	0.000429
3930	0.065903	0.673934	0.0	0.000067	0.000017	0.000017	0.735386	0.000433
3978	0.134356	0.338190	0.0	0.000067	0.000034	0.000034	0.930215	0.000537

344 rows × 13 columns

5. KMeans Clustering With Clusters '4'

In [86]:
```
Airline_Data_N[Airline_Data_N['K_Clusters']==4]
```

Out[86]:

ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_m

Kmeans Clustering Ends

DBSCAN

In [96]:
```
Airline_Data_N.head()
```

Out[96]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	F
0	0.000034	0.970414	0.0	0.000034	0.000034	0.000034	0.006000	0.000034	
1	0.000098	0.940209	0.0	0.000049	0.000049	0.000049	0.010504	0.000098	
2	0.000071	0.981113	0.0	0.000024	0.000024	0.000024	0.097817	0.000095	
3	0.000245	0.904428	0.0	0.000061	0.000061	0.000061	0.030605	0.000061	
4	0.000047	0.912226	0.0	0.000037	0.000009	0.000009	0.404078	0.000243	

In [98]:
```
del Airline_Data_N['K_Clusters']
```

In [99]:
```
Airline_Data_N
```

Out[99]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
0	0.000034	0.970414	0.0	0.000034	0.000034	0.000034	0.006000	0.000034
1	0.000098	0.940209	0.0	0.000049	0.000049	0.000049	0.010504	0.000098
2	0.000071	0.981113	0.0	0.000024	0.000024	0.000024	0.097817	0.000095
3	0.000245	0.904428	0.0	0.000061	0.000061	0.000061	0.030605	0.000061
4	0.000047	0.912226	0.0	0.000037	0.000009	0.000009	0.404078	0.000243
...	...	...	...	...	...	...	...	...
3994	0.193227	0.888739	0.0	0.000048	0.000048	0.000048	0.410072	0.000192
3995	0.062263	0.997710	0.0	0.000015	0.000015	0.000015	0.015202	0.000077
3996	0.051533	0.943692	0.0	0.000038	0.000013	0.000013	0.326292	0.000103
3997	0.073000	0.996925	0.0	0.000018	0.000018	0.000018	0.009080	0.000018
3998	0.770721	0.578089	0.0	0.000192	0.000192	0.000192	0.000000	0.000000

3999 rows × 12 columns

In [100]:
```
from sklearn.cluster import DBSCAN
```

In [101]:
```
Dbs = DBSCAN(min_samples=2, eps=0.2)
```

In [102]:
```
Dbs_Clusters = Dbs.fit_predict(Airline_Data_N)
Dbs_Clusters
```

Out[102]:
```
array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

In [103]:
```
Airline_Data_N['Dbs_Clusters'] = Dbs_Clusters
```

In [104]:
```
Airline_Data_N
```

Out[104]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
0	0.000034	0.970414	0.0	0.000034	0.000034	0.000034	0.006000	0.000034
1	0.000098	0.940209	0.0	0.000049	0.000049	0.000049	0.010504	0.000098
2	0.000071	0.981113	0.0	0.000024	0.000024	0.000024	0.097817	0.000095
3	0.000245	0.904428	0.0	0.000061	0.000061	0.000061	0.030605	0.000061
4	0.000047	0.912226	0.0	0.000037	0.000009	0.000009	0.404078	0.000243
...	...	...	...	...	...	...	...	...
3994	0.193227	0.888739	0.0	0.000048	0.000048	0.000048	0.410072	0.000192
3995	0.062263	0.997710	0.0	0.000015	0.000015	0.000015	0.015202	0.000077
3996	0.051533	0.943692	0.0	0.000038	0.000013	0.000013	0.326292	0.000103
3997	0.073000	0.996925	0.0	0.000018	0.000018	0.000018	0.009080	0.000018
3998	0.770721	0.578089	0.0	0.000192	0.000192	0.000192	0.000000	0.000000

3999 rows × 13 columns

1. DBSCAN With Clusters '0'

In [105]:
```
Airline_Data_N[Airline_Data_N['Dbs_Clusters']==0]
```

Out[105]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
0	0.000034	0.970414	0.0	0.000034	0.000034	0.000034	0.006000	0.000034
1	0.000098	0.940209	0.0	0.000049	0.000049	0.000049	0.010504	0.000098
2	0.000071	0.981113	0.0	0.000024	0.000024	0.000024	0.097817	0.000095
3	0.000245	0.904428	0.0	0.000061	0.000061	0.000061	0.030605	0.000061
4	0.000047	0.912226	0.0	0.000037	0.000009	0.000009	0.404078	0.000243
...	...	...	...	...	...	...	...	...
3994	0.193227	0.888739	0.0	0.000048	0.000048	0.000048	0.410072	0.000192
3995	0.062263	0.997710	0.0	0.000015	0.000015	0.000015	0.015202	0.000077
3996	0.051533	0.943692	0.0	0.000038	0.000013	0.000013	0.326292	0.000103
3997	0.073000	0.996925	0.0	0.000018	0.000018	0.000018	0.009080	0.000018
3998	0.770721	0.578089	0.0	0.000192	0.000192	0.000192	0.000000	0.000000

3975 rows × 13 columns

2. DBSCAN With Clusters '1'

In [106]:
```
Airline_Data_N[Airline_Data_N['Dbs_Clusters']==1]
```

Out[106]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
511	0.032251	0.611339	0.036743	0.000062	0.000062	0.000062	0.645525	0.002121
1146	0.096820	0.617290	0.000000	0.000084	0.000084	0.000084	0.604080	0.000920
1262	0.079545	0.717093	0.000000	0.000062	0.000062	0.000062	0.525623	0.000936
1990	0.159287	0.695797	0.000000	0.000079	0.000079	0.000079	0.522939	0.000635

3. DBSCAN With Clusters '2'

In [107]:
```
Airline_Data_N[Airline_Data_N['Dbs_Clusters']==2]
```

Out[107]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
1561	0.143567	0.816403	0.408201	0.000091	0.000091	0.000091	0.0	0.0
2142	0.121567	0.862963	0.431481	0.000056	0.000056	0.000056	0.0	0.0

4. DBSCAN With Clusters '3'

In [108]:
```
Airline_Data_N[Airline_Data_N['Dbs_Clusters']==3]
```

Out[108]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
2486	0.319082	0.568107	0.0	0.000127	0.000127	0.000127	0.603900	0.001783
2817	0.283509	0.709821	0.0	0.000100	0.000100	0.000100	0.559029	0.000799
3782	0.297904	0.627046	0.0	0.000078	0.000078	0.000078	0.634327	0.000940

5. DBSCAN With Clusters '4'

In [109]:
```
Airline_Data_N[Airline_Data_N['Dbs_Clusters']==4]
```

Out[109]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
2903	0.480596	0.476488	0.0	0.000164	0.000164	0.000164	0.476488	0.004765
3824	0.369484	0.528246	0.0	0.000096	0.000096	0.000096	0.528246	0.001056

6. DBSCAN With Clusters '5'

In [110]:
```
Airline_Data_N[Airline_Data_N['Dbs_Clusters']==5]
```

Out[110]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
3498	0.493466	0.134543	0.0	0.000140	0.000140	0.000140	0.840896	0.000140
3605	0.511764	0.037381	0.0	0.000141	0.000141	0.000141	0.846358	0.000141

7. 1. DBSCAN With Clusters '6'

In [111]:
```
Airline_Data_N[Airline_Data_N['Dbs_Clusters']==6]
```

Out[111]:

ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_m

DBSCAN ENDS

Final Conclusion:-

From all clustering methods DBSCAN is more optimized, Since it has more number of clusters as compared to Hierarchical & KMeans clustering