Clustering Assignment

Data Set : EastWestAirlines

1. Import Necessary libraries

In [1]:
```
import pandas as pd
```

2. Import Data

In [2]:
```
airline_details = pd.read_excel('EastWestAirlines.xlsx',sheet_name = 'data')
airline_details
```

Out[2]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?
0	1	28143	0	1	1	1	174	1	0	0	7000	0
1	2	19244	0	1	1	1	215	2	0	0	6968	0
2	3	41354	0	1	1	1	4123	4	0	0	7034	0
3	4	14776	0	1	1	1	500	1	0	0	6952	0
4	5	97752	0	4	1	1	43300	26	2077	4	6935	1
...	...	...	...	...	...	...	...	...	...	...	...	...
3994	4017	18476	0	1	1	1	8525	4	200	1	1403	1
3995	4018	64385	0	1	1	1	981	5	0	0	1395	1
3996	4019	73597	0	3	1	1	25447	8	0	0	1402	1
3997	4020	54899	0	1	1	1	500	1	500	1	1401	0
3998	4021	3016	0	1	1	1	0	0	0	0	1398	0

3999 rows × 12 columns

3. Data Understanding

In [3]:
```
airline_details.head()
```

Out[3]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?
0	1	28143	0	1	1	1	174	1	0	0	7000	0
1	2	19244	0	1	1	1	215	2	0	0	6968	0
2	3	41354	0	1	1	1	4123	4	0	0	7034	0
3	4	14776	0	1	1	1	500	1	0	0	6952	0
4	5	97752	0	4	1	1	43300	26	2077	4	6935	1

In [4]:
```
airline_details.shape
```

Out[4]:
```
(3999, 12)
```

In [5]:
```
airline_details.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3999 entries, 0 to 3998
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   ID#                3999 non-null   int64
 1   Balance            3999 non-null   int64
 2   Qual_miles         3999 non-null   int64
 3   cc1_miles          3999 non-null   int64
 4   cc2_miles          3999 non-null   int64
 5   cc3_miles          3999 non-null   int64
 6   Bonus_miles        3999 non-null   int64
 7   Bonus_trans        3999 non-null   int64
 8   Flight_miles_12mo  3999 non-null   int64
 9   Flight_trans_12    3999 non-null   int64
 10  Days_since_enroll  3999 non-null   int64
 11  Award?             3999 non-null   int64
dtypes: int64(12)
memory usage: 375.0 KB
```

In [6]:
```
airline_details.isna().sum()
```

Out[6]:
```
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 [7]:
```
airline_details.describe()
```

Out[7]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?
count	3999.000000	3.999000e+03	3999.000000	3999.000000	3999.000000	3999.000000	3999.000000	3999.00000	3999.000000	3999.000000	3999.00000	3999.000000
mean	2014.819455	7.360133e+04	144.114529	2.059515	1.014504	1.012253	17144.846212	11.60190	460.055764	1.373593	4118.55939	0.370343
std	1160.764358	1.007757e+05	773.663804	1.376919	0.147650	0.195241	24150.967826	9.60381	1400.209171	3.793172	2065.13454	0.482957
min	1.000000	0.000000e+00	0.000000	1.000000	1.000000	1.000000	0.000000	0.00000	0.000000	0.000000	2.00000	0.000000
25%	1010.500000	1.852750e+04	0.000000	1.000000	1.000000	1.000000	1250.000000	3.00000	0.000000	0.000000	2330.00000	0.000000
50%	2016.000000	4.309700e+04	0.000000	1.000000	1.000000	1.000000	7171.000000	12.00000	0.000000	0.000000	4096.00000	0.000000
75%	3020.500000	9.240400e+04	0.000000	3.000000	1.000000	1.000000	23800.500000	17.00000	311.000000	1.000000	5790.50000	1.000000
max	4021.000000	1.704838e+06	11148.000000	5.000000	3.000000	5.000000	263685.000000	86.00000	30817.000000	53.000000	8296.00000	1.000000

In [8]:
```
airline_details.dtypes
```

Out[8]:
```
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
```

4. CLUSTERING

(a) HIERARCHICAL CLUSTERING

STEP 1 = Data Pre-processing

In [9]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')
```

In [10]:
```
airline_details = pd.read_excel('EastWestAirlines.xlsx',sheet_name = 'data')
airline_details.head()
```

Out[10]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?
0	1	28143	0	1	1	1	174	1	0	0	7000	0
1	2	19244	0	1	1	1	215	2	0	0	6968	0
2	3	41354	0	1	1	1	4123	4	0	0	7034	0
3	4	14776	0	1	1	1	500	1	0	0	6952	0
4	5	97752	0	4	1	1	43300	26	2077	4	6935	1

In [11]:
```
from sklearn.preprocessing import MinMaxScaler
```

In [12]:
```
std_scaler_1 = MinMaxScaler()
x_1 = pd.DataFrame(std_scaler_1.fit_transform(airline_details.iloc[:,1:]))
x_1
```

Out[12]:

	0	1	2	3	4	5	6	7	8	9	10
0	0.016508	0.0	0.00	0.0	0.0	0.000660	0.011628	0.000000	0.000000	0.843742	0.0
1	0.011288	0.0	0.00	0.0	0.0	0.000815	0.023256	0.000000	0.000000	0.839884	0.0
2	0.024257	0.0	0.00	0.0	0.0	0.015636	0.046512	0.000000	0.000000	0.847842	0.0
3	0.008667	0.0	0.00	0.0	0.0	0.001896	0.011628	0.000000	0.000000	0.837955	0.0
4	0.057338	0.0	0.75	0.0	0.0	0.164211	0.302326	0.067398	0.075472	0.835905	1.0
...	...	...	...	...	...	...	...	...	...	...	...
3994	0.010837	0.0	0.00	0.0	0.0	0.032330	0.046512	0.006490	0.018868	0.168917	1.0
3995	0.037766	0.0	0.00	0.0	0.0	0.003720	0.058140	0.000000	0.000000	0.167953	1.0
3996	0.043169	0.0	0.50	0.0	0.0	0.096505	0.093023	0.000000	0.000000	0.168797	1.0
3997	0.032202	0.0	0.00	0.0	0.0	0.001896	0.011628	0.016225	0.018868	0.168676	0.0
3998	0.001769	0.0	0.00	0.0	0.0	0.000000	0.000000	0.000000	0.000000	0.168314	0.0

3999 rows × 11 columns

STEP 2 = Finding the optimal number of clusters using the Dendrogram :

In [13]:
```
from scipy.cluster.hierarchy import linkage
import scipy.cluster.hierarchy as sch
```

For creating dendrogram ::: p = np.array(df_norm) and converting into numpy array format

In [14]:
```
# create dendrogram
x = linkage(x_1, method = 'complete',metric = 'euclidean')

plt.figure(figsize = (15, 5))
plt.title('Hierarchical Dendrogram Plot')
plt.xlabel('Index')
plt.ylabel('Euclidean Distances')

sch.dendrogram(x)
plt.show()
```

STEP 3 = Training the hierarchical clustering model :

In [15]:
```
from sklearn.cluster import AgglomerativeClustering
```

In [16]:
```
hc = AgglomerativeClustering(n_clusters = 5, linkage = 'complete',affinity = "euclidean").fit(x_1)
hc
```

Out[16]:
```
AgglomerativeClustering(linkage='complete', n_clusters=5)
```

In [17]:
```
x_hc = hc.fit_predict(x_1)
x_hc
```

Out[17]:
```
array([0, 0, 0, ..., 2, 0, 0], dtype=int64)
```

In [18]:
```
add_Clusters = pd.DataFrame(x_hc,columns = ['Clusters_1'])
add_Clusters.head()
```

Out[18]:

	Clusters_1
0	0
1	0
2	0
3	0
4	1

In [19]:
```
airline_details['clusters_1'] = add_Clusters
airline_details.head()
```

Out[19]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?	clusters_1
0	1	28143	0	1	1	1	174	1	0	0	7000	0	0
1	2	19244	0	1	1	1	215	2	0	0	6968	0	0
2	3	41354	0	1	1	1	4123	4	0	0	7034	0	0
3	4	14776	0	1	1	1	500	1	0	0	6952	0	0
4	5	97752	0	4	1	1	43300	26	2077	4	6935	1	1

STEP 4 = Visualizing the clusters :

In [20]:
```
airline_details[airline_details['clusters_1']==1]
```

Out[20]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?	clusters_1
4	5	97752	0	4	1	1	43300	26	2077	4	6935	1	1
11	12	96522	0	5	1	1	61105	19	0	0	6924	1	1
15	16	28495	0	4	1	1	49442	15	0	0	6912	1	1
16	17	51890	0	4	1	1	48963	16	0	0	6910	1	1
20	21	120576	0	5	1	1	58831	23	250	2	6896	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
3753	3776	70178	0	5	1	1	61530	12	1300	6	3306	1	1
3772	3795	822321	0	5	1	1	138334	26	600	2	3288	1	1
3846	3869	97510	1678	5	1	1	71609	51	7650	26	1665	1	1
3855	3878	190730	0	5	1	1	78916	28	2450	9	1648	1	1
3883	3906	126167	0	5	1	1	190160	15	0	0	1612	1	1

325 rows × 13 columns

In [21]:
```
airline_details[airline_details['clusters_1']==2]
```

Out[21]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?	clusters_1
7	8	20856	0	1	1	1	5250	4	250	1	6938	1	2
8	9	443003	0	3	2	1	1753	43	3850	12	6948	1	2
9	10	104860	0	3	1	1	28426	28	1150	3	6931	1	2
17	18	13958	0	1	1	1	4291	5	0	0	6905	1	2
18	19	91473	0	3	1	1	27408	17	0	0	6903	1	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...
3987	4010	11933	0	1	1	1	249	3	79	1	1412	1	2
3989	4012	2622	0	1	1	1	1625	6	0	0	1404	1	2
3994	4017	18476	0	1	1	1	8525	4	200	1	1403	1	2
3995	4018	64385	0	1	1	1	981	5	0	0	1395	1	2
3996	4019	73597	0	3	1	1	25447	8	0	0	1402	1	2

1144 rows × 13 columns

In [22]:
```
airline_details[airline_details['clusters_1']==3]
```

Out[22]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?	clusters_1
2015	2032	53232	888	4	1	1	80696	65	22100	45	3831	1	3
3235	3257	287033	0	1	1	1	26161	58	12873	53	2272	1	3
3583	3606	160114	500	1	1	1	71954	86	30817	53	1373	1	3
3594	3617	27619	0	4	1	1	83726	68	14050	46	1325	1	3

(b) K - MEANS

STEP 1 = Data Pre-processing :

In [23]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')
```

In [24]:
```
airline_details = pd.read_excel('EastWestAirlines.xlsx',sheet_name = 'data')
airline_details.head()
```

Out[24]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?
0	1	28143	0	1	1	1	174	1	0	0	7000	0
1	2	19244	0	1	1	1	215	2	0	0	6968	0
2	3	41354	0	1	1	1	4123	4	0	0	7034	0
3	4	14776	0	1	1	1	500	1	0	0	6952	0
4	5	97752	0	4	1	1	43300	26	2077	4	6935	1

In [25]:
```
from sklearn.preprocessing import StandardScaler
```

In [26]:
```
std_scaler_2 = StandardScaler()
x_2 = std_scaler_2.fit_transform(airline_details.iloc[:,1:])
print(x_2)
```

```
[[-4.51140783e-01 -1.86298687e-01 -7.69578406e-01 ... -3.62167870e-01
   1.39545434e+00 -7.66919299e-01]
 [-5.39456874e-01 -1.86298687e-01 -7.69578406e-01 ... -3.62167870e-01
   1.37995704e+00 -7.66919299e-01]
 [-3.20031232e-01 -1.86298687e-01 -7.69578406e-01 ... -3.62167870e-01
   1.41192021e+00 -7.66919299e-01]
 ...
 [-4.29480975e-05 -1.86298687e-01  6.83121167e-01 ... -3.62167870e-01
  -1.31560393e+00  1.30391816e+00]
 [-1.85606976e-01 -1.86298687e-01 -7.69578406e-01 ... -9.85033311e-02
  -1.31608822e+00 -7.66919299e-01]
 [-7.00507951e-01 -1.86298687e-01 -7.69578406e-01 ... -3.62167870e-01
  -1.31754109e+00 -7.66919299e-01]]
```

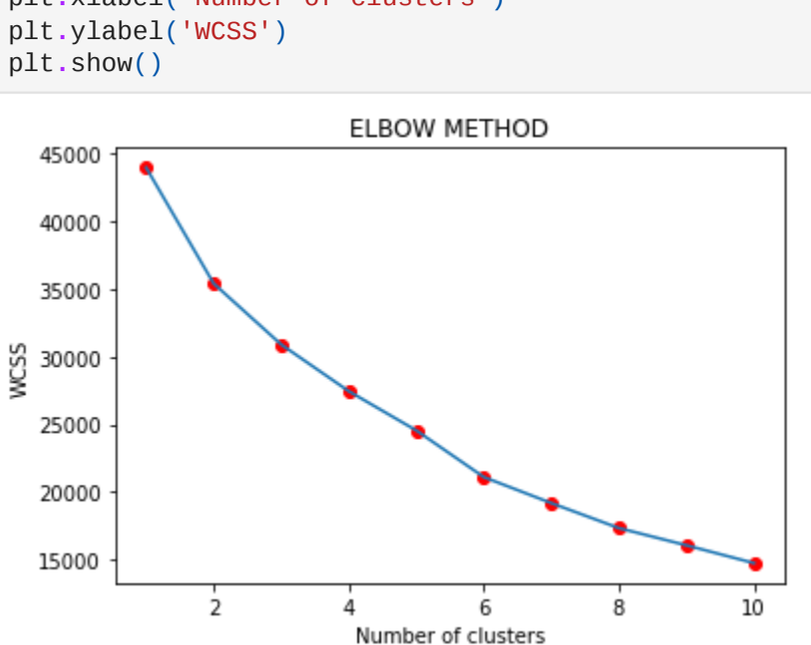
STEP 2 = Finding the optimal number of clusters using the Elbow Method :¶

In [27]:
```
from matplotlib import pyplot as plt
from sklearn.cluster import KMeans
```

In [28]:
```
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i,random_state = 0)
    kmeans.fit(x_2)
    wcss.append(kmeans.inertia_)
wcss
```

Out[28]:
```
[43989.000000000015,
 35409.51145242436,
 30899.246824652197,
 27446.37245668138,
 24541.32958204578,
 21089.973628929645,
 19165.861458655665,
 17316.866446128297,
 16043.406242309518,
 14726.983322766795]
```

In [29]:
```
plt.plot(range(1, 11), wcss)
plt.scatter(range(1, 11), wcss,color = 'red')
plt.title('ELBOW METHOD')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



STEP 3 = Training the K - means clustering model :

In [30]:
```
km = KMeans(6, random_state = 40)
km.fit(x_1)
```

Out[30]:
```
KMeans(n_clusters=6, random_state=40)
```

In [31]:
```
km.labels_
```

Out[31]:
```
array([4, 4, 4, ..., 1, 0, 0])
```

In [32]:
```
airline_details['clusters_2'] = km.labels_
km.cluster_centers_
```

Out[32]:
```
array([[ 1.94137515e-02,  8.44382696e-03,  1.76841085e-02,
         8.23643411e-03,  1.69573643e-03,  1.23045313e-02,
         7.17842978e-02,  6.90692719e-03,  1.13902296e-02,
         2.39980966e-01, -4.44089210e-16],
       [ 4.00536989e-02,  1.52189249e-02,  6.31853786e-01,
         6.93889390e-18,  6.52741514e-04,  1.23136025e-01,
         2.17651345e-01,  1.82155063e-02,  3.21198089e-02,
         4.03697410e-01,  1.00000000e+00],
       [ 6.93891884e-02,  6.59020789e-03,  6.46035599e-01,
         8.09061489e-04,  5.66343042e-03,  1.19022293e-01,
         2.00383834e-01,  7.27197078e-03,  1.18458814e-02,
         5.32620376e-01, -3.88578059e-16],
       [ 4.89953609e-02,  2.60542873e-02,  3.90044577e-02,
         1.63447251e-02,  2.22882615e-03,  3.35642727e-02,
         1.21825219e-01,  3.34267751e-02,  5.94073285e-02,
         5.22892182e-01,  1.00000000e+00],
       [ 2.92823328e-02,  8.06451613e-03,  3.05299539e-02,
         9.79262673e-03,  2.88018433e-04,  1.31485480e-02,
         8.03906334e-02,  7.91232441e-03,  1.37379358e-02,
         6.71078504e-01, -4.99600361e-16],
       [ 8.46963593e-02,  2.01093311e-02,  8.16470588e-01,
         1.17647059e-03,  1.17647059e-02,  2.17879658e-01,
         2.50451436e-01,  2.76163080e-02,  4.79023307e-02,
         7.50551639e-01,  1.00000000e+00]])
```

In [33]:
```
airline_details.groupby('clusters_2').agg(['mean']).reset_index()
```

Out[33]:

	clusters_2	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?
		mean	mean	mean	mean	mean	mean	mean	mean	mean	mean	mean	mean
0	0	3204.917636	33097.301357	94.131783	1.070736	1.016473	1.006783	3244.520349	6.173450	212.850775	0.603682	1992.402132	0.0
1	1	2456.846753	68025.958442	168.779221	3.524675	1.000000	1.002597	32376.987013	18.688312	558.431169	1.693506	3364.901299	1.0
2	2	1840.462783	118297.325243	73.467638	3.584142	1.001618	1.022654	31384.393204	17.233010	224.100324	0.627832	4419.553398	0.0
3	3	1904.763744	83529.153046	290.453195	1.156018	1.032689	1.008915	8850.395245	10.476969	1030.112927	3.148588	4338.867756	1.0
4	4	1183.362903	49921.633641	89.903226	1.122120	1.019585	1.001152	3467.074885	6.913594	243.834101	0.728111	5567.925115	0.0
5	5	844.990544	144989.255319	225.238771	4.271868	1.002364	1.047281	57653.576832	21.579196	855.075650	2.550827	6227.356974	1.0

STEP 4 = Visualizing the clusters :

In [34]:
```
airline_details[airline_details['clusters_2']==1]
```

Out[34]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?	clusters_2
647	656	25835	0	3	1	1	15153	15	0	0	6142	1	1
759	769	43179	0	3	1	1	11915	6	0	0	6214	1	1
768	778	10978	0	3	1	1	14313	11	0	0	6193	1	1
803	813	66122	0	3	1	1	11756	13	0	0	5630	1	1
805	815	42044	2253	3	1	1	11433	10	0	0	6140	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
3931	3954	49519	0	3	1	1	25762	18	0	0	1522	1	1
3954	3977	33972	0	3	1	1	21029	29	0	0	1467	1	1
3973	3996	72297	0	3	1	1	16241	16	0	0	1587	1	1
3979	4002	57793	0	3	1	1	20959	15	1198	3	1424	1	1
3996	4019	73597	0	3	1	1	25447	8	0	0	1402	1	1

385 rows × 13 columns

In [35]:
```
airline_details[airline_details['clusters_2']==2]
```

Out[35]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?	clusters_2
6	7	84914	0	3	1	1	27482	25	0	0	6994	0	2
30	31	177926	0	3	1	1	20797	13	0	0	7886	0	2
32	33	276571	0	4	1	1	42044	23	0	0	7872	0	2
44	45	121260	0	3	1	1	18493	18	0	0	7808	0	2
56	57	75971	0	4	1	1	34339	14	0	0	6869	0	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...
3966	3989	91605	0	4	1	1	37840	18	0	0	1444	0	2
3974	3997	58387	0	3	1	1	14581	39	0	0	1435	0	2
3980	4003	28867	0	3	1	1	19169	28	0	0	1425	0	2
3985	4008	59017	0	4	1	1	34746	25	0	0	1418	0	2
3991	4014	39142	0	3	1	1	14981	28	0	0	1408	0	2

618 rows × 13 columns

In [36]:
```
airline_details[airline_details['clusters_2']==3]
```

Out[36]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?	clusters_2
7	8	20856	0	1	1	1	5250	4	250	1	6938	1	3
17	18	13958	0	1	1	1	4291	5	0	0	6905	1	3
21	22	185681	2024	1	1	1	13300	16	1800	9	6896	1	3
29	30	19221	0	1	1	1	4655	8	500	1	7896	1	3
36	37	23193	0	1	1	1	650	4	550	3	7861	1	3
...	...	...	...	...	...	...	...	...	...	...	...	...	...
3978	4001	10071	0	2	1	1	27701	16	0	0	1421	1	3
3987	4010	11933	0	1	1	1	249	3	79	1	1412	1	3
3989	4012	2622	0	1	1	1	1625	6	0	0	1404	1	3
3994	4017	18476	0	1	1	1	8525	4	200	1	1403	1	3
3995	4018	64385	0	1	1	1	981	5	0	0	1395	1	3

673 rows × 13 columns

(c) DBSCAN

STEP 1 = Data Pre-processing :

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
airline_details = pd.read_excel('EastWestAirlines.xlsx',sheet_name = 'data')
airline_details.head()
```

Out[2]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?
0	1	28143	0	1	1	1	174	1	0	0	7000	0
1	2	19244	0	1	1	1	215	2	0	0	6968	0
2	3	41354	0	1	1	1	4123	4	0	0	7034	0
3	4	14776	0	1	1	1	500	1	0	0	6952	0
4	5	97752	0	4	1	1	43300	26	2077	4	6935	1

In [3]:
```
df = pd.DataFrame(airline_details)
df.head()
```

Out[3]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?
0	1	28143	0	1	1	1	174	1	0	0	7000	0
1	2	19244	0	1	1	1	215	2	0	0	6968	0
2	3	41354	0	1	1	1	4123	4	0	0	7034	0
3	4	14776	0	1	1	1	500	1	0	0	6952	0
4	5	97752	0	4	1	1	43300	26	2077	4	6935	1

In [4]:
```
df.columns
```

Out[4]:
```
Index(['ID#', 'Balance', 'Qual_miles', 'cc1_miles', 'cc2_miles', 'cc3_miles',
       'Bonus_miles', 'Bonus_trans', 'Flight_miles_12mo', 'Flight_trans_12',
       'Days_since_enroll', 'Award?'],
      dtype='object')
```

In [5]:
```
df1 = df.drop(['cc1_miles', 'cc2_miles', 'cc3_miles','ID#','Award?'],axis = 1)
df1.head()
```

Out[5]:

	Balance	Qual_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll
0	28143	0	174	1	0	0	7000
1	19244	0	215	2	0	0	6968
2	41354	0	4123	4	0	0	7034
3	14776	0	500	1	0	0	6952
4	97752	0	43300	26	2077	4	6935

STEP 2 = Finding the optimal number of clusters using the DBSCAN :

In [6]:
```
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler
```

In [7]:
```
std_scaler_3 = StandardScaler().fit(df1)
x_3 = std_scaler_3.transform(df1)
x_3
```

Out[7]:
```
array([[-4.51140783e-01, -1.86298687e-01, -7.02786266e-01, ...,
        -3.28603258e-01, -3.62167870e-01,  1.39545434e+00],
       [-5.39456874e-01, -1.86298687e-01, -7.01088399e-01, ...,
        -3.28603258e-01, -3.62167870e-01,  1.37995704e+00],
       [-3.20031232e-01, -1.86298687e-01, -5.39252702e-01, ...,
        -3.28603258e-01, -3.62167870e-01,  1.41192021e+00],
       ...,
       [-4.29480975e-05, -1.86298687e-01,  3.43803696e-01, ...,
        -3.28603258e-01, -3.62167870e-01, -1.31560393e+00],
       [-1.85606976e-01, -1.86298687e-01, -6.89286154e-01, ...,
         2.85309024e-02, -9.85033311e-02, -1.31608822e+00],
       [-7.00507951e-01, -1.86298687e-01, -7.09991847e-01, ...,
        -3.28603258e-01, -3.62167870e-01, -1.31754109e+00]])
```

STEP 3 = Training the DBSCAN clustering model :

In [8]:
```
db = DBSCAN(eps = 0.45,min_samples = 5)
db.fit(x_3)
```

Out[8]:
```
DBSCAN(eps=0.45)
```

In [9]:
```
db.labels_
```

Out[9]:
```
array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

In [10]:
```
df2 = pd.DataFrame(db.labels_,columns = ['clusters_3'])
df2.head()
```

Out[10]:

	clusters_3
0	0
1	0
2	0
3	0
4	-1

In [11]:
```
airline_details['clusters_3'] = df2['clusters_3']
airline_details.head()
```

Out[11]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?	clusters_3
0	1	28143	0	1	1	1	174	1	0	0	7000	0	0
1	2	19244	0	1	1	1	215	2	0	0	6968	0	0
2	3	41354	0	1	1	1	4123	4	0	0	7034	0	0
3	4	14776	0	1	1	1	500	1	0	0	6952	0	0
4	5	97752	0	4	1	1	43300	26	2077	4	6935	1	-1

In [12]:
```
df2.clusters_3.value_counts()
```

Out[12]:
```
 0    2741
-1    1220
 5       7
 8       6
 1       5
 4       5
 6       5
 2       4
 3       3
 7       3
Name: clusters_3, dtype: int64
```

In [13]:
```
airline_details.groupby(['clusters_3']).agg(['mean']).reset_index()
```

Out[13]:

	clusters_3	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?
		mean	mean	mean	mean	mean	mean	mean	mean	mean	mean	mean	mean
0	-1	1709.103279	141698.390164	472.388525	2.767213	1.013934	1.036066	33592.175410	18.470492	1372.644262	3.995082	4711.558197	0.601639
1	0	2141.523532	42575.058738	0.000000	1.739876	1.014958	1.001824	9778.432324	8.504196	56.954396	0.212696	3870.742795	0.267421
2	1	1931.800000	77792.400000	0.000000	3.800000	1.000000	1.000000	32710.400000	13.800000	510.000000	1.200000	4124.600000	0.600000
3	2	1591.500000	174606.000000	0.000000	4.750000	1.000000	1.000000	56291.000000	22.000000	0.000000	0.000000	4919.750000	0.500000
4	3	1759.333333	20466.333333	0.000000	1.000000	1.000000	1.000000	966.666667	5.333333	966.666667	5.333333	4524.000000	0.666667
5	4	2406.800000	200722.200000	0.000000	4.000000	1.000000	1.000000	42755.600000	19.400000	0.000000	0.000000	3474.000000	0.000000
6	5	3083.428571	129873.142857	0.000000	1.000000	1.000000	1.000000	2657.142857	3.142857	485.714286	1.857143	1932.857143	0.571429
7	6	3334.000000	175683.800000	0.000000	1.000000	1.000000	1.000000	7961.800000	4.000000	0.000000	0.000000	1987.200000	0.600000
8	7	2820.666667	176415.666667	0.000000	1.000000	1.000000	1.000000	5481.333333	5.666667	58.333333	0.333333	3128.333333	0.000000
9	8	3698.000000	48710.500000	0.000000	2.500000	1.000000	1.000000	16142.666667	37.166667	0.000000	0.000000	1369.500000	0.000000

STEP 4 = Visualizing the clusters :

In [14]:
```
airline_details[airline_details['clusters_3']==3]
```

Out[14]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?	clusters_3
1636	1652	31230	0	1	1	1	815	5	815	5	4931	0	3
1794	1811	17609	0	1	1	1	803	5	803	5	4317	1	3
1798	1815	12560	0	1	1	1	1282	6	1282	6	4324	1	3

In [15]:
```
airline_details[airline_details['clusters_3']==2]
```

Out[15]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?	clusters_3
1441	1455	161321	0	5	1	1	59645	21	0	0	5147	0	2
1525	1540	156219	0	5	1	1	55874	24	0	0	5015	1	2
1619	1634	196079	0	5	1	1	56890	21	0	0	4791	0	2
1721	1737	184805	0	4	1	1	52755	22	0	0	4726	1	2

In [16]:
```
airline_details[airline_details['clusters_3']==1]
```

Out[16]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?	clusters_3
1519	1534	71930	0	4	1	1	32774	13	500	1	4300	1	1
1800	1817	86143	0	3	1	1	29652	11	500	1	4317	1	1
1817	1834	75112	0	4	1	1	35532	14	500	1	4271	0	1
2123	2140	75505	0	4	1	1	30850	17	500	1	4007	0	1
2316	2334	80272	0	4	1	1	34744	14	550	2	3728	1	1