Performing DBSCAN clustering method on Airlines

```
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
data = pd.read_excel('/content/EastWestAirlines.xlsx', sheet_name='data')
```

```
data.head()
```

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trai
0	1	28143	0	1	1	1	174	
1	2	19244	0	1	1	1	215	
2	3	41354	0	1	1	1	4123	
3	4	14776	0	1	1	1	500	
4	5	97752	0	4	1	1	43300	2

```
data.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
```

```
data.drop(['ID#'], axis=1, inplace=True)
```

```
data.head()
```

	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
0	28143	0	1	1	1	174	1
1	19244	0	1	1	1	215	2
2	41354	0	1	1	1	4123	4
3	14776	0	1	1	1	500	1
4	97752	0	4	1	1	43300	26

```
array = data.values
```

```
array
```

```
array([[28143,     0,     1, ...,     0,  7000,     0],
       [19244,     0,     1, ...,     0,  6968,     0],
       [41354,     0,     1, ...,     0,  7034,     0],
       ...,
       [73597,     0,     3, ...,     0,  1402,     1],
       [54899,     0,     1, ...,     1,  1401,     0],
       [ 3016,     0,     1, ...,     0,  1398,     0]])
```

```
#Normalization method
stscaler = StandardScaler().fit(array)
X = stscaler.transform(array)
```

```
X
```

```
array([[-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]])
```

```
dbscan = DBSCAN(eps=0.8, min_samples=13)
dbscan.fit(X)
```

```
DBSCAN(algorithm='auto', eps=0.8, leaf_size=30, metric='euclidean',
       metric_params=None, min_samples=13, n_jobs=None, p=None)
```

```
#Noisy sample are given the lable. -1
dbscan.labels_
```

```
array([0, 0, 0, ..., 1, 0, 0])
```

```
cl = pd.DataFrame(dbscan.labels_, columns=['cluster'])
```

```
cl
```

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

3999 rows × 1 columns

```
cl.value_counts()
```

```
cluster
 0         2129
-1          980
 1          815
 2           51
 4           12
 3           12
dtype: int64
```

```
pd.concat([data,cl], axis=1)
```

	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_tra
0	28143	0	1	1	1	174	
1	19244	0	1	1	1	215	
2	41354	0	1	1	1	4123	
3	14776	0	1	1	1	500	
4	97752	0	4	1	1	43300	

there are 980 outliers in the data and most of the data lies in 0 cluster.

3996	73597	0	3	1	1	25447
3998	3016	0	1	1	1	0

3999 rows × 12 columns

✓ 0s completed at 2:59 AM