In [28]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
import scipy.cluster.hierarchy as sch
from sklearn.cluster import AgglomerativeClustering
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler
```

In [14]:
```
Airline = pd.read_csv('EastWestAirlines.csv')
Airline
```

Out[14]:

	ID#	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Fli
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	

3999 rows × 12 columns

In [15]:
```
Airline.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 [33]:
```
airline2=Airline.drop(['ID#'],axis=1)
airline2
```

Out[33]:

	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_mil
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	
...	...	...	...	...	...	...	...	
3994	18476	0	1	1	1	8525	4	
3995	64385	0	1	1	1	981	5	
3996	73597	0	3	1	1	25447	8	
3997	54899	0	1	1	1	500	1	
3998	3016	0	1	1	1	0	0	

3999 rows × 12 columns

In [22]:
```
# Normalize heterogenous numerical data using z-score (x-mean/std) or custom defi
# Normalization function - here custom defined
def norm_func(i):
    x = (i-i.min())/(i.max()-i.min())
    return (x)
```

In [23]:
```
# Normalized data frame (considering the numerical part of data)
airline2_norm = norm_func(airline2)
airline2_norm
```

Out[23]:

	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_mi
0	0.016508	0.0	0.00	0.0	0.0	0.000660	0.011628	
1	0.011288	0.0	0.00	0.0	0.0	0.000815	0.023256	
2	0.024257	0.0	0.00	0.0	0.0	0.015636	0.046512	
3	0.008667	0.0	0.00	0.0	0.0	0.001896	0.011628	
4	0.057338	0.0	0.75	0.0	0.0	0.164211	0.302326	
...	...	...	...	...	...	...	...	
3994	0.010837	0.0	0.00	0.0	0.0	0.032330	0.046512	
3995	0.037766	0.0	0.00	0.0	0.0	0.003720	0.058140	
3996	0.043169	0.0	0.50	0.0	0.0	0.096505	0.093023	
3997	0.032202	0.0	0.00	0.0	0.0	0.001896	0.011628	
3998	0.001769	0.0	0.00	0.0	0.0	0.000000	0.000000	

3999 rows × 11 columns

In [29]:
```
# Normalize heterogenous numerical data using standard scalar fit transform to da
airline3_norm=StandardScaler().fit_transform(Airline)
airline3_norm
```

Out[29]:
```
array([[-1.73512503e+00, -4.51140783e-01, -1.86298687e-01, ...,
        -3.62167870e-01,  1.39545434e+00, -7.66919299e-01],
       [-1.73426342e+00, -5.39456874e-01, -1.86298687e-01, ...,
        -3.62167870e-01,  1.37995704e+00, -7.66919299e-01],
       [-1.73340181e+00, -3.20031232e-01, -1.86298687e-01, ...,
        -3.62167870e-01,  1.41192021e+00, -7.66919299e-01],
       ...,
       [ 1.72682006e+00, -4.29480975e-05, -1.86298687e-01, ...,
        -3.62167870e-01, -1.31560393e+00,  1.30391816e+00],
       [ 1.72768167e+00, -1.85606976e-01, -1.86298687e-01, ...,
        -9.85033311e-02, -1.31608822e+00, -7.66919299e-01],
       [ 1.72854328e+00, -7.00507951e-01, -1.86298687e-01, ...,
        -3.62167870e-01, -1.31754109e+00, -7.66919299e-01]])
```

In [27]:
```
# DBSCAN Clustering
dbscan = DBSCAN(eps=1,min_samples=4)
dbscan.fit(airline2_norm)
```

Out[27]:
```
DBSCAN(eps=1, min_samples=4)
```

In [30]:
```python
#Noisy samples are given the label -1.
dbscan.labels_
```

Out[30]:
```
array([0, 0, 0, ..., 1, 0, 0], dtype=int64)
```

In [36]:
```python
# Adding clusters to dataset
airline2['clusters']=dbscan.labels_
airline2
```

Out[36]:

	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_mil
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	
...	...	...	...	...	...	...	...	
3994	18476	0	1	1	1	8525	4	
3995	64385	0	1	1	1	981	5	
3996	73597	0	3	1	1	25447	8	
3997	54899	0	1	1	1	500	1	
3998	3016	0	1	1	1	0	0	

3999 rows × 12 columns

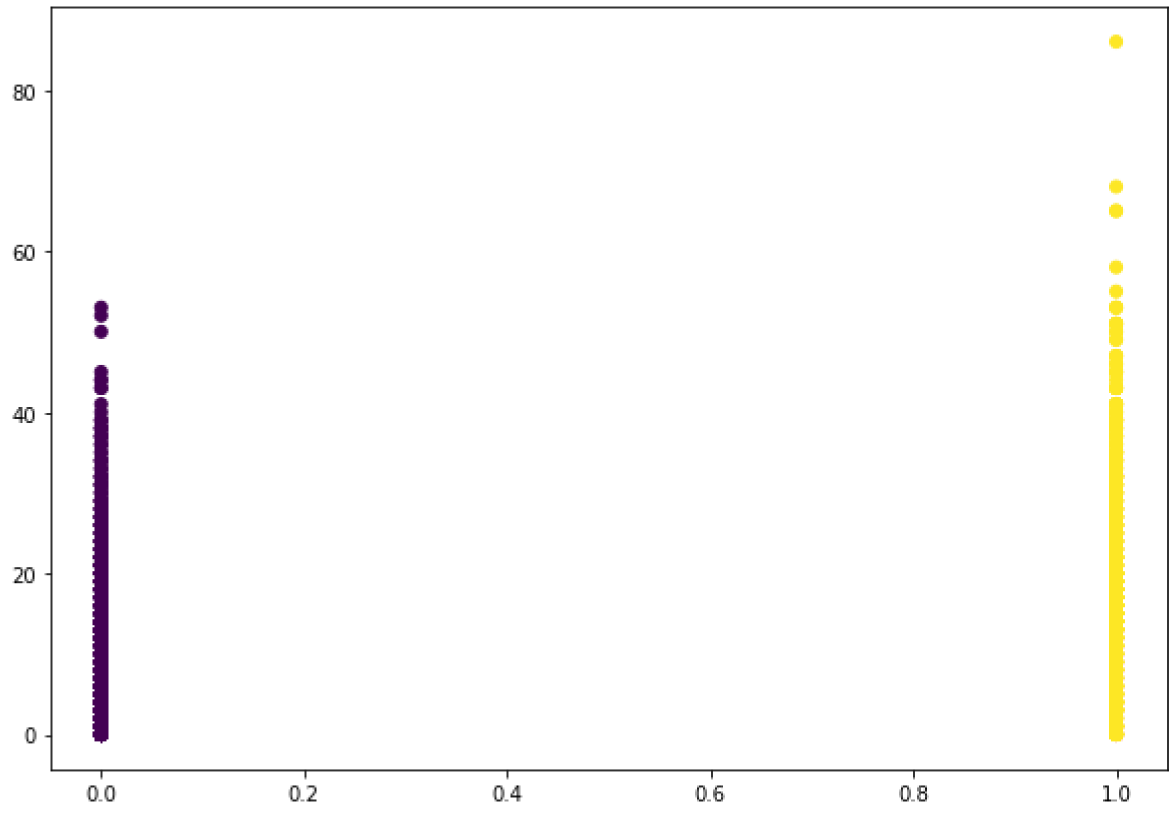
In [37]:
```python
airline2.groupby('clusters').agg(['mean']).reset_index()
```

Out[37]:

clusters	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Fli
	mean	mean	mean	mean	mean	mean	mean	me
0	59807.839555	87.602462	1.705322	1.01390	1.008737	10227.689039	9.142971	
1	97053.051317	240.196489	2.661715	1.01553	1.018231	28905.414585	15.782579	

In [42]:
```
# Plot Clusters
plt.figure(figsize=(10, 7))
plt.scatter(airline2['clusters'],airline2['Bonus_trans'], c=dbscan.labels_)
```

Out[42]:
```
<matplotlib.collections.PathCollection at 0x2ad1c0ce790>
```



In [ ]: