In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
import scipy.cluster.hierarchy as sch
from sklearn.cluster import AgglomerativeClustering
from sklearn.preprocessing import normalize
```

In [3]:
```
airlines=pd.read_csv('EastWestAirlines_csv.csv')
airlines
```

Out[3]:

	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 [4]:
```
airlines.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 [5]:
```
airlines.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]:
```
airlines.shape
```

Out[6]:
```
(3999, 12)
```

In [7]:
```
airlines.isna().sum()
```

Out[7]:
```
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
```

Hierarchical clustering

In [61]:
```
airlines_2 = airlines.drop(['ID#'], axis = 1)
airlines_2
```

Out[61]:

	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

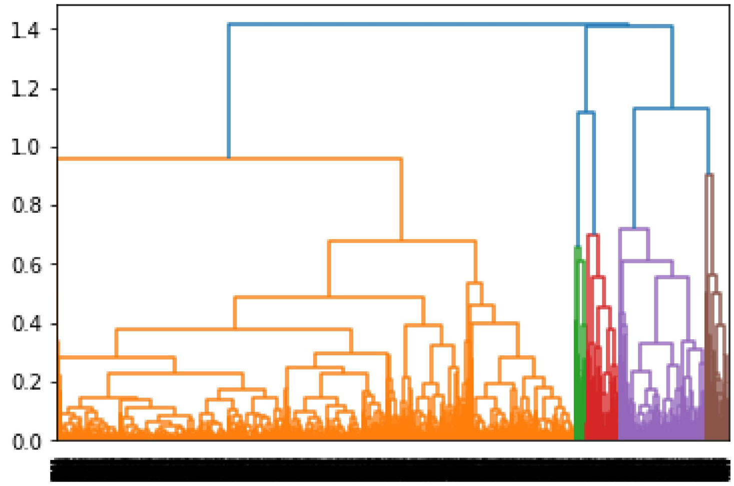
In [9]:
```
# Normalize Heterogenous numerical data
airlines_2_norm = pd.DataFrame(normalize(airlines_2),columns=airlines_2.columns)
airlines_2_norm
```

Out[9]:

	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_m
0	0.970414	0.0	0.000034	0.000034	0.000034	0.006000	0.000034	
1	0.940209	0.0	0.000049	0.000049	0.000049	0.010504	0.000098	
2	0.981113	0.0	0.000024	0.000024	0.000024	0.097817	0.000095	
3	0.904428	0.0	0.000061	0.000061	0.000061	0.030605	0.000061	
4	0.912226	0.0	0.000037	0.000009	0.000009	0.404078	0.000243	
...	...	...	...	...	...	...	...	
3994	0.905810	0.0	0.000049	0.000049	0.000049	0.417949	0.000196	
3995	0.999649	0.0	0.000016	0.000016	0.000016	0.015231	0.000078	
3996	0.944948	0.0	0.000039	0.000013	0.000013	0.326726	0.000103	
3997	0.999592	0.0	0.000018	0.000018	0.000018	0.009104	0.000018	
3998	0.907271	0.0	0.000301	0.000301	0.000301	0.000000	0.000000	

3999 rows × 11 columns

In [10]:
```
# Create Dendrogram
dendrograms = sch.dendrogram(sch.linkage(airlines_2_norm,'complete'))
```



In [11]:
```
# Create Clusters
hclusters = AgglomerativeClustering(n_clusters=5, affinity='euclidean', linkage='
hclusters
```

Out[11]: AgglomerativeClustering(n_clusters=5)

In [12]:
```
# Save Clsuters for Chart
y_hc = hclusters.fit_predict(airlines_2_norm)
y_hc
```

Out[12]: array([4, 2, 2, ..., 2, 4, 2], dtype=int32)

In [13]:
```
clusters=pd.DataFrame(y_hc, columns=['clusters'])
clusters
```

Out[13]:

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

3999 rows × 1 columns

In [14]:
```
airlines_2['clusters'] = clusters
airlines_2
```

Out[14]:

	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 [15]:
```
airlines_2[airlines_2['clusters']==0]
```

Out[15]:

	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_mil
27	8828	0	1	1	1	0	0	
31	10021	0	1	1	1	0	0	
39	2176	0	1	1	1	0	0	
51	1300	0	1	1	1	370	1	
55	14448	0	1	1	1	1625	6	
...	...	...	...	...	...	...	...	
3861	3126	0	1	1	1	100	1	
3876	1000	0	1	1	1	0	0	
3942	2131	0	1	1	1	405	3	
3981	1010	0	1	1	1	0	0	
3984	404	0	1	1	1	550	3	

229 rows × 12 columns

In [16]:
```
airlines_2[airlines_2['clusters']==1]
```

Out[16]:

	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_mil
15	28495	0	4	1	1	49442	15	
16	51890	0	4	1	1	48963	16	
41	10470	0	4	1	1	38094	26	
58	38077	0	3	1	1	34024	8	
78	49238	0	4	1	1	38037	18	
...	...	...	...	...	...	...	...	
3919	5000	0	1	1	1	5000	1	
3924	14775	0	1	1	1	14275	9	
3930	40424	0	4	1	1	44110	26	
3944	2124	0	1	1	1	2324	2	
3978	10071	0	2	1	1	27701	16	

453 rows × 12 columns

In [17]:
```
airlines_2[airlines_2['clusters']==2]
```

Out[17]:

	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_mil
1	19244	0	1	1	1	215	2	
2	41354	0	1	1	1	4123	4	
3	14776	0	1	1	1	500	1	
5	16420	0	1	1	1	0	0	
6	84914	0	3	1	1	27482	25	
...	...	...	...	...	...	...	...	
3989	2622	0	1	1	1	1625	6	
3992	11181	0	1	1	1	929	12	
3993	3974	0	1	1	1	365	3	
3996	73597	0	3	1	1	25447	8	
3998	3016	0	1	1	1	0	0	

1547 rows × 12 columns

In [18]:
```
airlines_2[airlines_2['clusters']==3]
```

Out[18]:

	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_
4	97752	0	4	1	1	43300	26	
11	96522	0	5	1	1	61105	19	
19	23354	0	3	1	1	10447	5	
20	120576	0	5	1	1	58831	23	
28	59763	0	3	1	1	33772	20	
...	...	...	...	...	...	...	...	
3986	34235	0	1	1	1	18910	7	
3988	5000	0	1	1	1	2125	3	
3990	11310	0	1	1	1	5021	2	
3991	39142	0	3	1	1	14981	28	
3994	18476	0	1	1	1	8525	4	

In [19]:
```
airlines_2[airlines_2['clusters']==4]
```

Out[19]:

	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_mil
0	28143	0	1	1	1	174	1	
8	443003	0	3	2	1	1753	43	
21	185681	2024	1	1	1	13300	16	
23	66275	0	1	1	1	2533	11	
24	205651	500	1	1	1	4025	21	
...	...	...	...	...	...	...	...	
3982	11463	0	1	1	1	339	4	
3983	26173	0	1	1	1	305	1	
3987	11933	0	1	1	1	249	3	
3995	64385	0	1	1	1	981	5	
3997	54899	0	1	1	1	500	1	

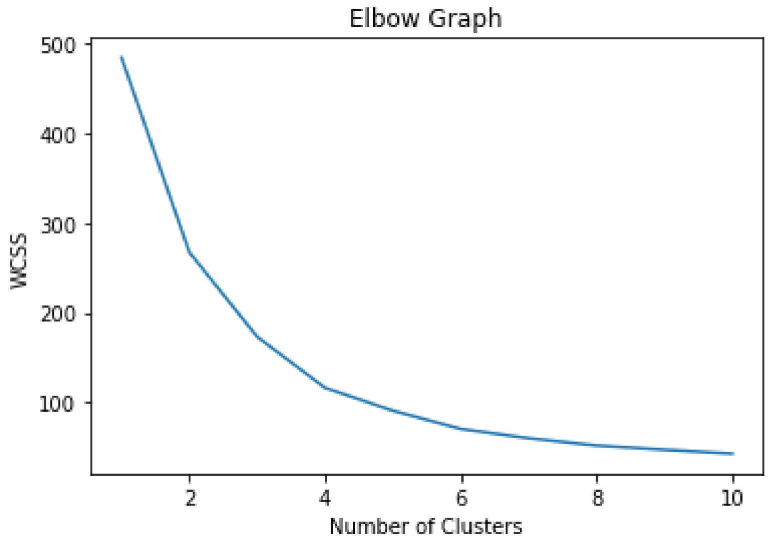
1191 rows × 12 columns

K-Mean Clustering

In [20]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [24]:
```
wcss=[]
for i in range(1,11):
    kmeans= KMeans(n_clusters=i, random_state=2)
    kmeans.fit(airlines_2_norm)
    wcss.append(kmeans.inertia_)
```

In [25]:
```
plt.plot(range(1,11), wcss)
plt.title('Elbow Graph')
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.show()
```



In [27]:
```
# Build Cluster algorithm using K=4
clusters4=KMeans(4,random_state=30).fit(airlines_2_norm)
clusters4
```

Out[27]:
```
KMeans(n_clusters=4, random_state=30)
```

In [28]:
```
clusters4.labels_
```

Out[28]:
```
array([3, 3, 3, ..., 0, 3, 3])
```

In [30]:
```
# Assign clusters to the data set
airlines4=airlines_2.copy()
airlines4['clusters4id']=clusters4.labels_
airlines4
```

Out[30]:

	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 × 13 columns

In [31]:
```
#compute the centroids for K=4 clusters with 11 variables
clusters4.cluster_centers_
```

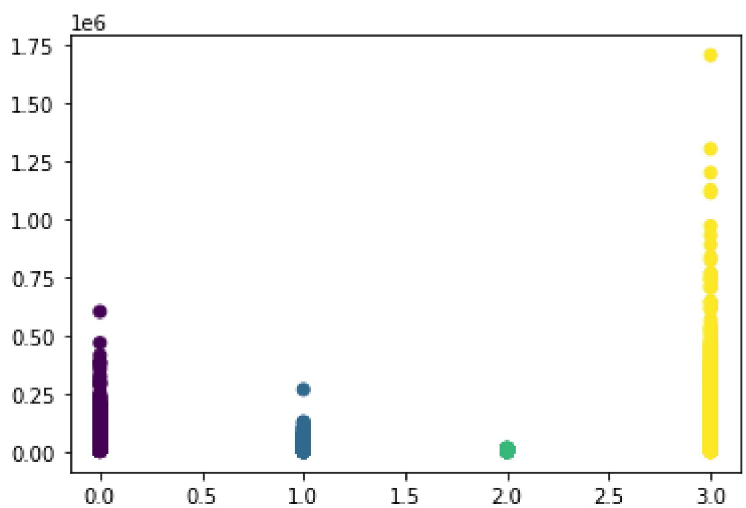
Out[31]:
```
array([[8.99048678e-01, 2.03403471e-03, 5.68074076e-05, 3.01913199e-05,
        2.95156437e-05, 4.03089039e-01, 4.02398112e-04, 7.62262675e-03,
        2.24052643e-05, 8.50654942e-02, 9.73901648e-06],
       [5.23653977e-01, 2.37603195e-03, 9.13653056e-05, 4.56081254e-05,
        4.45095230e-05, 7.97866700e-01, 5.07019477e-04, 1.75075997e-02,
        5.89123100e-05, 1.31443994e-01, 3.00837174e-05],
       [6.28081328e-01, 9.30359261e-04, 2.06331617e-04, 2.06128767e-04,
        2.05879951e-04, 1.23980626e-01, 4.76413717e-04, 6.66146530e-03,
        2.24385615e-05, 6.89106611e-01, 2.58980762e-05],
       [9.82878899e-01, 3.71612347e-03, 4.15057209e-05, 3.77179195e-05,
        3.76205578e-05, 8.06914054e-02, 1.57453088e-04, 6.65079627e-03,
        2.12921781e-05, 1.03324885e-01, 4.81770304e-06]])
```

In [32]:
```
# Group data by Clusters K=4
airlines4.groupby('clusters4id').agg(['mean']).reset_index()
```

Out[32]:

	clusters4id	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_tr
		mean	mean	mean	mean	mean	mean	m
0	0	72378.903670	119.606422	3.077982	1.024771	1.018349	31486.477982	17.476
1	1	28617.579670	112.000000	3.280220	1.030220	1.068681	42166.565934	17.634
2	2	5129.247934	8.285124	1.004132	1.004132	1.000000	891.388430	3.012
3	3	88484.857577	175.062961	1.495441	1.008250	1.001737	8110.131568	8.770

In [34]:
```
plt.scatter(airlines4['clusters4id'],airlines4['Balance'],c=clusters4.labels_)
plt.show()
```



In [ ]:

In [35]:
```
# Build Cluster algorithm using K=5
clusters5=KMeans(5,random_state=30).fit(airlines_2_norm)
clusters5
```

Out[35]: KMeans(n_clusters=5, random_state=30)

In [36]:
```
clusters5.labels_
```

Out[36]: array([0, 4, 0, ..., 1, 0, 4])

In [37]:
```
# Assign clusters to the data set
airlines5=airlines_2.copy()
airlines5['clusters5id']=clusters5.labels_
airlines5
```

Out[37]:

	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 × 13 columns

In [38]:
```
#compute the centroids for K=4 clusters with 11 variables
clusters5.cluster_centers_
```

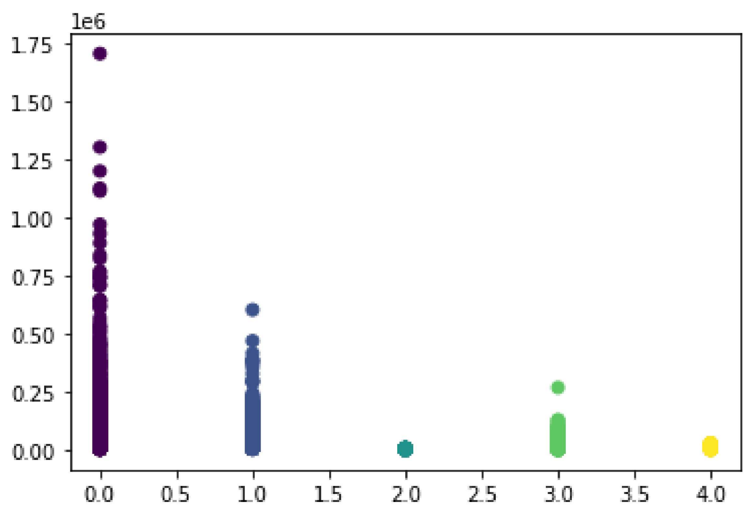
Out[38]:
```
array([[9.87336213e-01, 3.41678871e-03, 3.52693557e-05, 3.03417203e-05,
        3.02288024e-05, 9.16328114e-02, 1.54857161e-04, 6.61411635e-03,
        2.08307943e-05, 7.54405080e-02, 3.98926957e-06],
       [8.90453898e-01, 1.91306896e-03, 5.81394027e-05, 3.02384249e-05,
        2.95149925e-05, 4.23750290e-01, 4.07503085e-04, 7.83124032e-03,
        2.30666627e-05, 8.31457802e-02, 1.00567454e-05],
       [4.14644791e-01, 1.30104261e-18, 2.28611980e-04, 2.27627266e-04,
        2.27627266e-04, 1.50766683e-01, 5.97513433e-04, 7.35401490e-03,
        2.84888383e-05, 8.48268382e-01, 3.91049405e-05],
       [5.14097044e-01, 2.46403313e-03, 9.56772813e-05, 5.01782621e-05,
        4.88674224e-05, 8.02764990e-01, 5.20805294e-04, 1.79689628e-02,
        6.06455235e-05, 1.36723853e-01, 3.06681430e-05],
       [8.92936852e-01, 4.46454511e-03, 1.23968035e-04, 1.23783403e-04,
        1.23783403e-04, 7.58365867e-02, 2.93996886e-04, 6.32105922e-03,
        2.08016784e-05, 4.07924096e-01, 1.35510886e-05]])
```

In [39]:
```
# Group data by Clusters K=4
airlines5.groupby('clusters5id').agg(['mean']).reset_index()
```

Out[39]:

	clusters5id	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_tr
		mean	mean	mean	mean	mean	mean	m
0	0	97052.708990	187.518999	1.611214	1.009268	1.001854	9665.601946	9.722
1	1	71002.722782	110.376008	3.144153	1.026210	1.020161	32818.490927	17.717
2	2	2415.576577	0.000000	1.009009	1.000000	1.000000	850.189189	3.036
3	3	27462.797721	116.148148	3.245014	1.034188	1.071225	41806.162393	17.572
4	4	11756.307494	55.263566	1.005168	1.000000	1.000000	980.863049	3.444

In [40]:
```
plt.scatter(airlines5['clusters5id'],airlines5['Balance'],c=clusters5.labels_)
plt.show()
```



In [ ]:

DBSCAN Clustering

In [57]:
```
from sklearn.cluster import DBSCAN
```

In [58]:
```
dbscan = DBSCAN(eps=1,min_samples=2)
dbscan.fit(airlines_2_norm)
```

Out[58]: DBSCAN(eps=1, min_samples=2)

In [63]:
```
airlines_2['clusters']=dbscan.labels_
airlines_2
```

Out[63]:

	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	

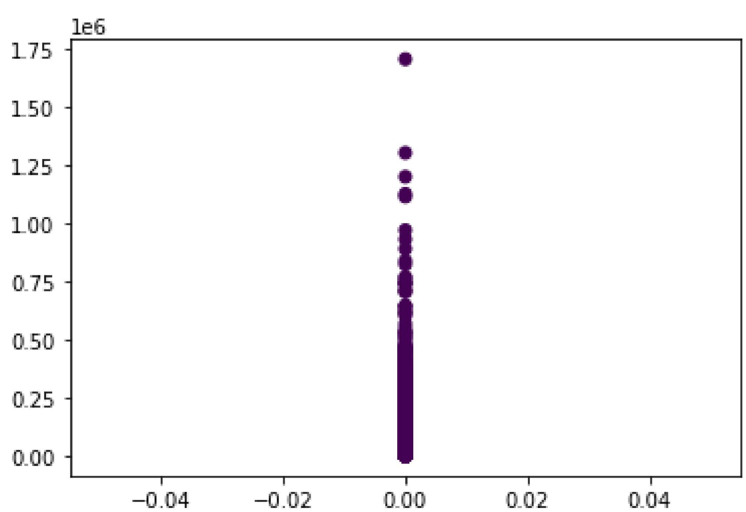
3999 rows × 12 columns

In [64]:
```
airlines_2.groupby('clusters').agg(['mean']).reset_index()
```

Out[64]:

	clusters	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
		mean	mean	mean	mean	mean	mean	mear
0	0	73601.327582	144.114529	2.059515	1.014504	1.012253	17144.846212	11.601

In [65]:
```
plt.scatter(airlines_2['clusters'],airlines_2['Balance'], c=dbscan.labels_)
plt.show()
```



In [ ]: