Question A:

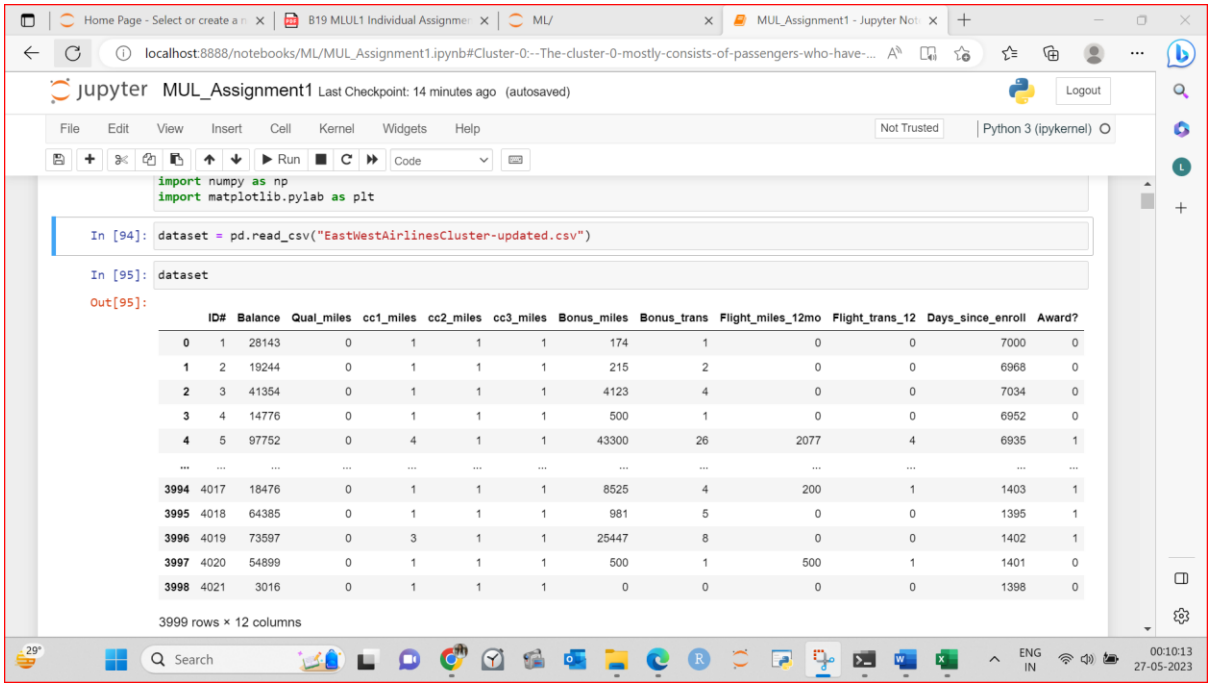
Yes, it is recommended to standardize the given data before applying clustering techniques because the standardization converges the variables to a similar scale by transforming them to have mean =0 and standard deviation = 1. This is important to ensure that no single feature dominates the distance or similarity calculations. In our dataset, variables have units in mileage and in terms of no. of transactions. So, while computing the distance formula we may not be able to arrive at correct formulated result representing the distance between two data points as variables with larger magnitudes will have significant influence. So, normalizing/ standardizing data to a similar scale becomes important.

Question B:

To perform hierarchical clustering with Euclidean distance and Ward's method, I have performed following steps:

1. Loaded the dataset and dropped #ID column as it won't be helpful in clustering. This column represents passenger id representing unique customers. I have considered rest of the columns for clustering as according to me they each have significance in deciding the offer: - For Example: -
    - Total balance miles = Balance + Qualifying miles + miles accrued through 3 credit cards in past 12 months + Bonus miles. So, each of these variables has a significant role in deciding the activity of the passengers. As airline wants to award passengers some mileage offers, it would be necessary for the airlines to undertake passengers' mileage accrual activity.
    - Bonus transaction can also help us understand the spending activity of the customer. I can find out average Bonus miles accrued per non flight transactions and then compare with the miles accrued from the 3 different credit cards. This will help me analyse segments of customer who are attracted to the non-flight offers and help me understand what offers they are mostly attracted to. This can either help me offer similar offers to the customer on my flyer credit card or help me develop better offers with the partners in the partnership program.
    - Flight miles and Flight transactions can help us understand if the passenger in this frequent flyer program is how frequent a flyer is and whether he is a long-distance traveller or not. Addition of these fields in deciding the clusters will help us to arrange an offer that can be made in accordance with international and domestic travellers.
    - Award: - If airline wants to promote the usage of its credit cards by providing limited offer period during summer vacations for example, then it would be useful for airlines to understand how many passengers have award flights. Because if there is a significant no. of passengers who have awarded flights then running such a campaign won't be successful. So, considering this factor in the clustering.

2. Conducted basic EDA with the loaded dataset to check any missing values/Null values.



jupyter MUL_Assignment1 Last Checkpoint: 14 minutes ago (autosaved)

```
import numpy as np
import matplotlib.pylab as plt
```

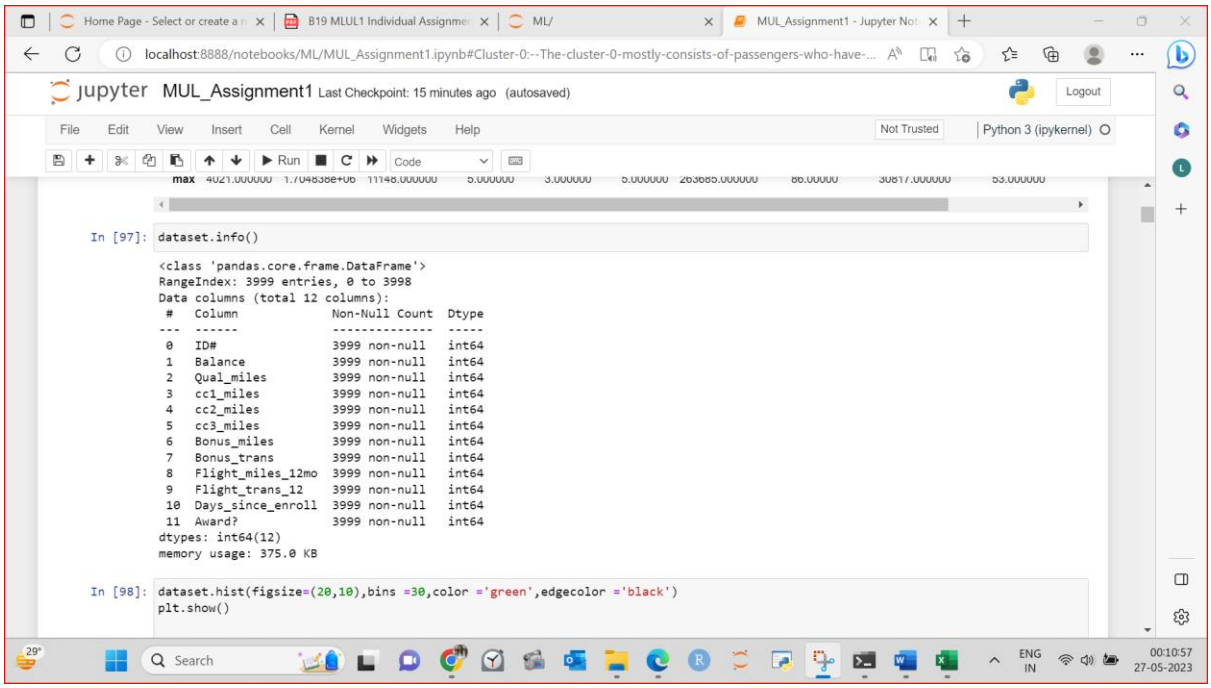
In [94]:
```
dataset = pd.read_csv("EastWestAirlinesCluster-updated.csv")
```

In [95]:
```
dataset
```

Out[95]:

	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



jupyter MUL_Assignment1 Last Checkpoint: 15 minutes ago (autosaved)

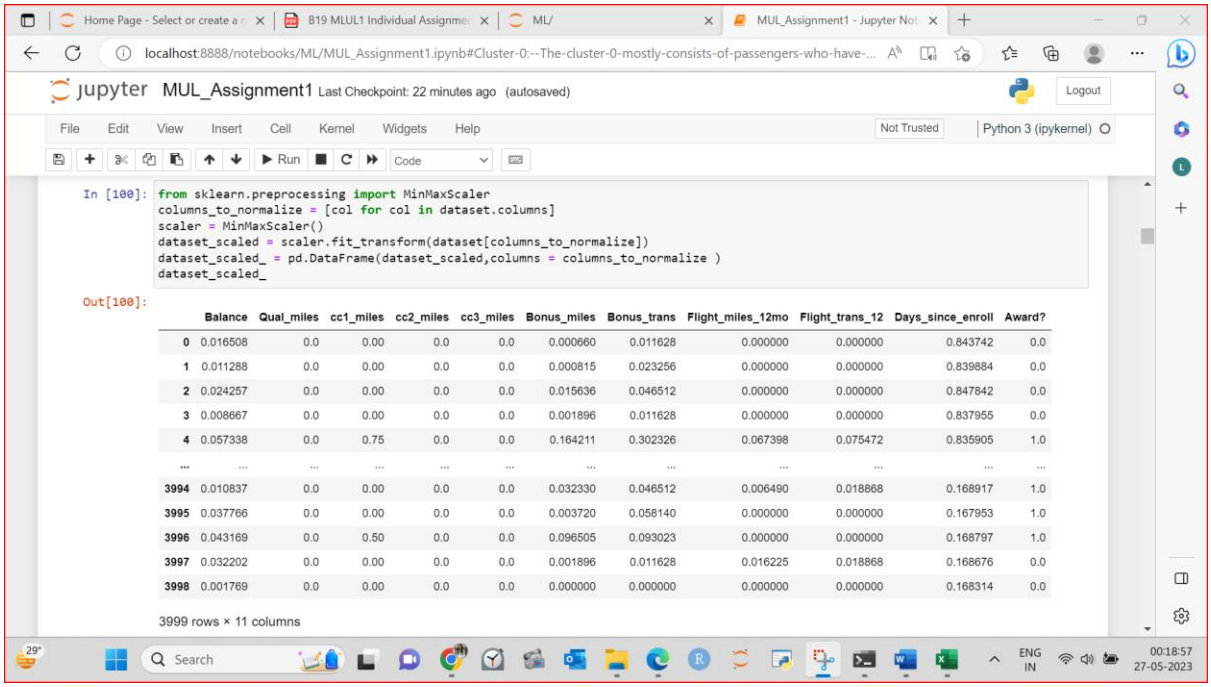
In [97]:
```
dataset.info()
```

```
<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 [98]:
```
dataset.hist(figsize=(20,10),bins =30,color ='green',edgecolor ='black')
plt.show()
```

3. To perform hierarchical clustering, I have standardized the data as we had data of different scales. I have used both the techniques Minmax Scaler and StandardScaler as I wanted to check whether different standardization has any effect on the clustering. Upon analysing, I have found that both the techniques provide same clustering and output represented by these scaled data is also the same.

A. MinMax Scaling:-



jupyter MUL_Assignment1 Last Checkpoint: 22 minutes ago (autosaved)

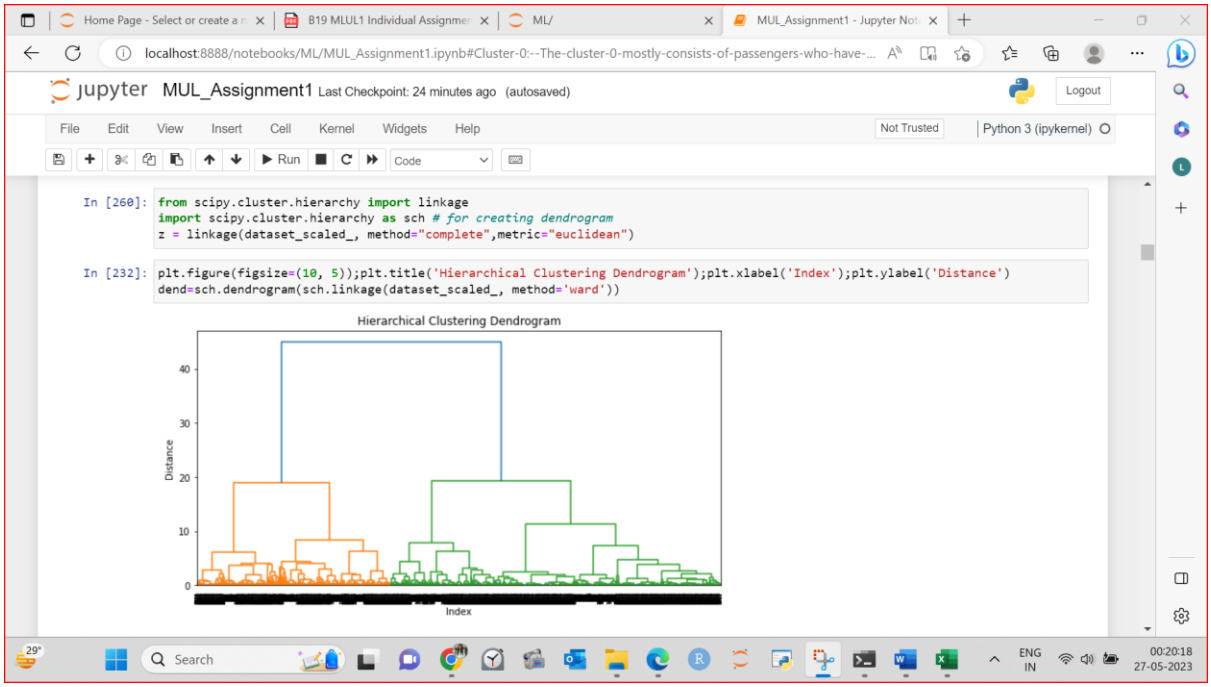
File Edit View Insert Cell Kernel Widgets Help — Not Trusted — Python 3 (ipykernel)

```
In [100]: from sklearn.preprocessing import MinMaxScaler
columns_to_normalize = [col for col in dataset.columns]
scaler = MinMaxScaler()
dataset_scaled = scaler.fit_transform(dataset[columns_to_normalize])
dataset_scaled_ = pd.DataFrame(dataset_scaled,columns = columns_to_normalize )
dataset_scaled_
```

Out[100]:

	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?
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



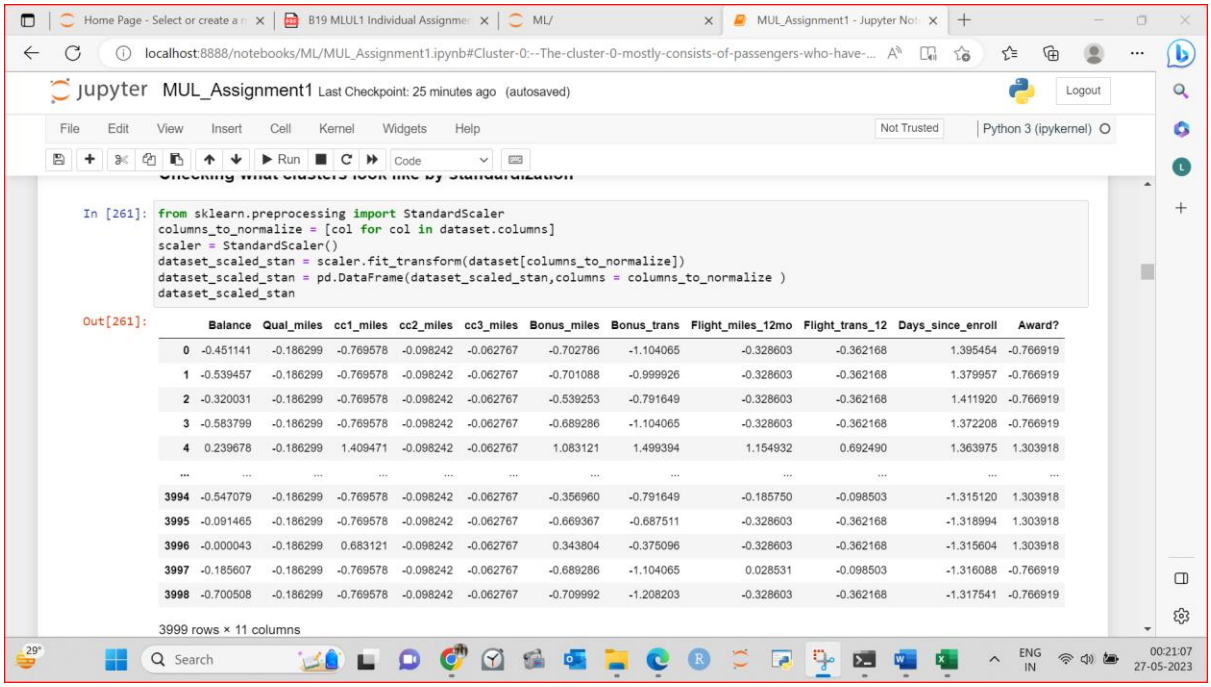
jupyter MUL_Assignment1 Last Checkpoint: 24 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help — Not Trusted — Python 3 (ipykernel)

```
In [260]: from scipy.cluster.hierarchy import linkage
import scipy.cluster.hierarchy as sch # for creating dendrogram
z = linkage(dataset_scaled_, method="complete",metric="euclidean")
```

```
In [232]: plt.figure(figsize=(10, 5));plt.title('Hierarchical Clustering Dendrogram');plt.xlabel('Index');plt.ylabel('Distance')
dend=sch.dendrogram(sch.linkage(dataset_scaled_, method='ward'))
```

B. Standard Scaling:-

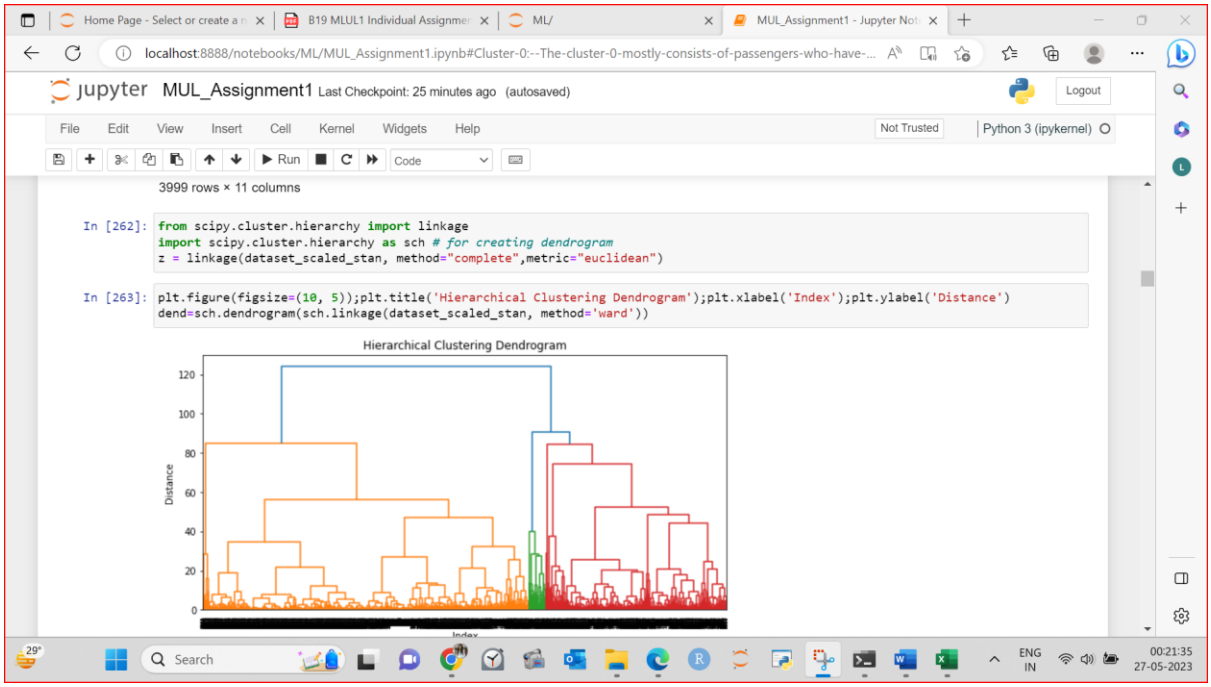


In [261]:
```
from sklearn.preprocessing import StandardScaler
columns_to_normalize = [col for col in dataset.columns]
scaler = StandardScaler()
dataset_scaled_stan = scaler.fit_transform(dataset[columns_to_normalize])
dataset_scaled_stan = pd.DataFrame(dataset_scaled_stan,columns = columns_to_normalize )
dataset_scaled_stan
```

Out[261]:

	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?
0	-0.451141	-0.186299	-0.769578	-0.098242	-0.062767	-0.702786	-1.104065	-0.328603	-0.362168	1.395454	-0.766919
1	-0.539457	-0.186299	-0.769578	-0.098242	-0.062767	-0.701088	-0.999926	-0.328603	-0.362168	1.379957	-0.766919
2	-0.320031	-0.186299	-0.769578	-0.098242	-0.062767	-0.539253	-0.791649	-0.328603	-0.362168	1.411920	-0.766919
3	-0.583799	-0.186299	-0.769578	-0.098242	-0.062767	-0.689286	-1.104065	-0.328603	-0.362168	1.372208	-0.766919
4	0.239678	-0.186299	1.409471	-0.098242	-0.062767	1.083121	1.499394	1.154932	0.692490	1.363975	1.303918
...	...	...	...	...	...	...	...	...	...	...	...
3994	-0.547079	-0.186299	-0.769578	-0.098242	-0.062767	-0.356960	-0.791649	-0.185750	-0.098503	-1.315120	1.303918
3995	-0.091465	-0.186299	-0.769578	-0.098242	-0.062767	-0.669367	-0.687511	-0.328603	-0.362168	-1.318994	1.303918
3996	-0.000043	-0.186299	0.683121	-0.098242	-0.062767	0.343804	-0.375096	-0.328603	-0.362168	-1.315604	1.303918
3997	-0.185607	-0.186299	-0.769578	-0.098242	-0.062767	-0.689286	-1.104065	0.028531	-0.098503	-1.316088	-0.766919
3998	-0.700508	-0.186299	-0.769578	-0.098242	-0.062767	-0.709992	-1.208203	-0.328603	-0.362168	-1.317541	-0.766919

3999 rows × 11 columns

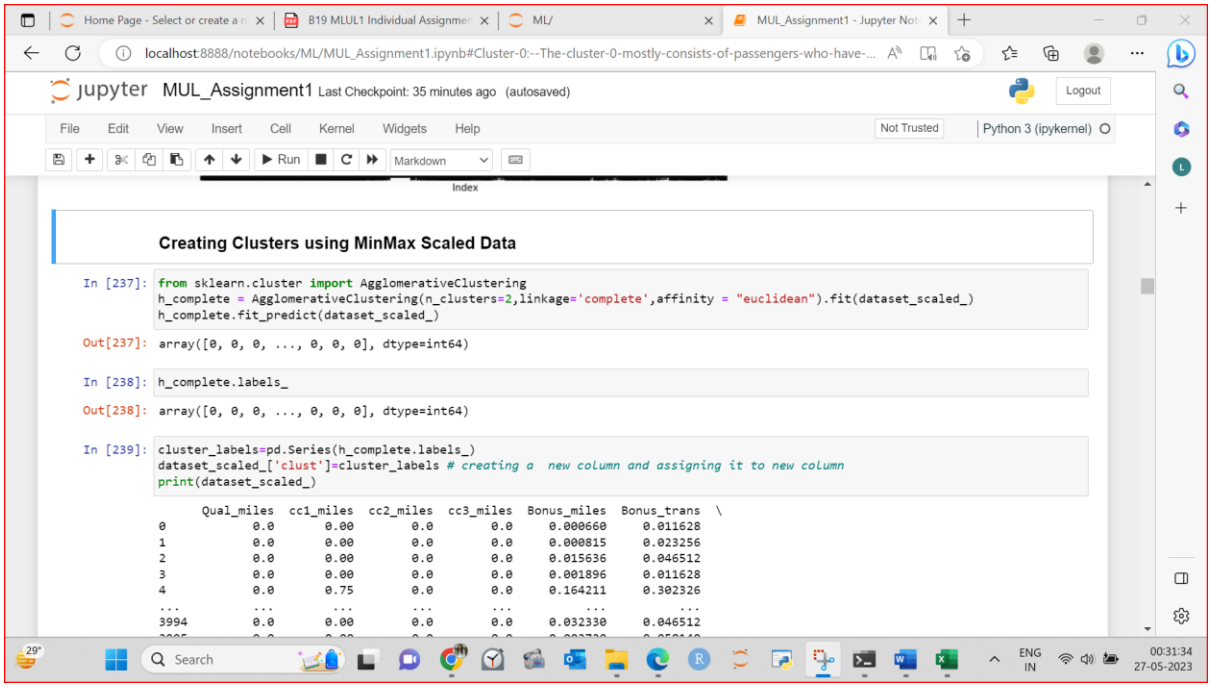


3999 rows × 11 columns

In [262]:
```
from scipy.cluster.hierarchy import linkage
import scipy.cluster.hierarchy as sch # for creating dendrogram
z = linkage(dataset_scaled_stan, method="complete",metric="euclidean")
```

In [263]:
```
plt.figure(figsize=(10, 5));plt.title('Hierarchical Clustering Dendrogram');plt.xlabel('Index');plt.ylabel('Distance')
dend=sch.dendrogram(sch.linkage(dataset_scaled_stan, method='ward'))
```

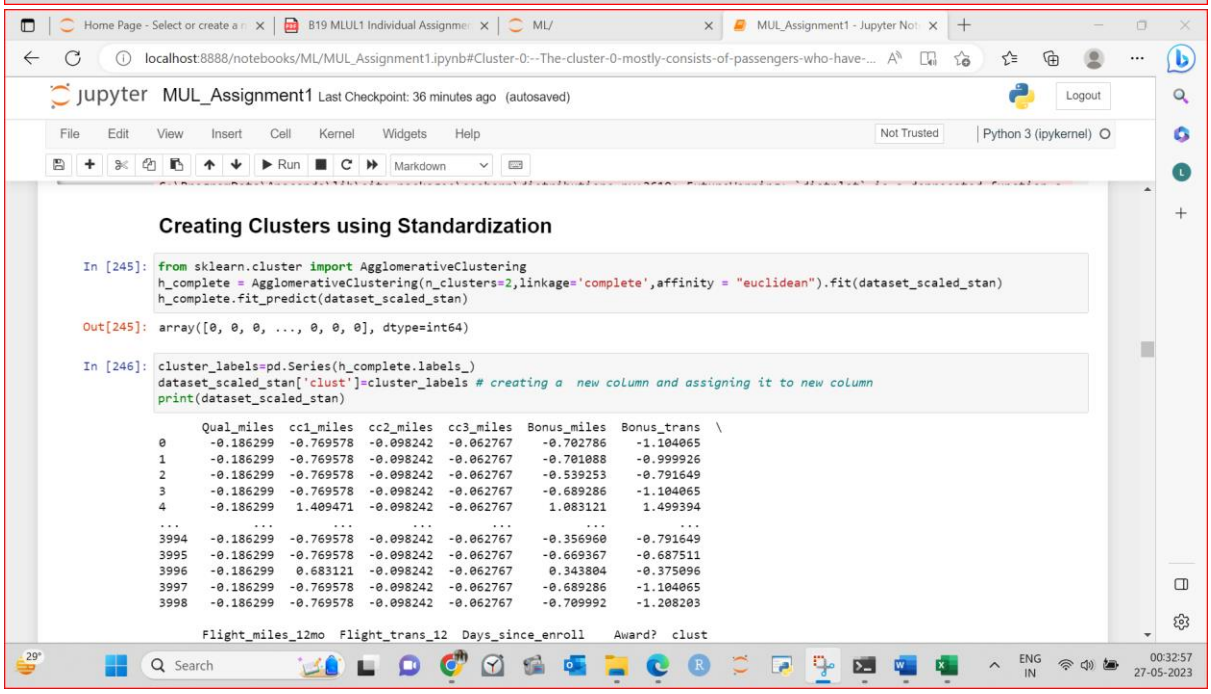
4. Upon analysing Denodagram made using both MinMax Scaler and Standard Scaler, I have come to a conclusion to cluster the dataset into 2 clusters. Now creating cluster on Standard data and MinMax normalized data and same clustering is applied on the main dataset to group the original data into clusters. Initially clustering labels used is 0 and 1. Later analysing data in both clusters using distplot as well as using excel, (The output is saved in **Assignment_1b.csv file**) I have come to following conclusions:-



Creating Clusters using MinMax Scaled Data

```
In [237]: from sklearn.cluster import AgglomerativeClustering
          h_complete = AgglomerativeClustering(n_clusters=2,linkage='complete',affinity = "euclidean").fit(dataset_scaled_)
          h_complete.fit_predict(dataset_scaled_)

Out[237]: array([0, 0, 0, ..., 0, 0, 0], dtype=int64)

In [238]: h_complete.labels_

Out[238]: array([0, 0, 0, ..., 0, 0, 0], dtype=int64)

In [239]: cluster_labels=pd.Series(h_complete.labels_)
          dataset_scaled_['clust']=cluster_labels # creating a  new column and assigning it to new column
          print(dataset_scaled_)
```

	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
0	0.0	0.00	0.0	0.0	0.000660	0.011628
1	0.0	0.00	0.0	0.0	0.000815	0.023256
2	0.0	0.00	0.0	0.0	0.015636	0.046512
3	0.0	0.00	0.0	0.0	0.001896	0.011628
4	0.0	0.75	0.0	0.0	0.164211	0.302326
...	...	...	...	...	...	...
3994	0.0	0.00	0.0	0.0	0.032330	0.046512



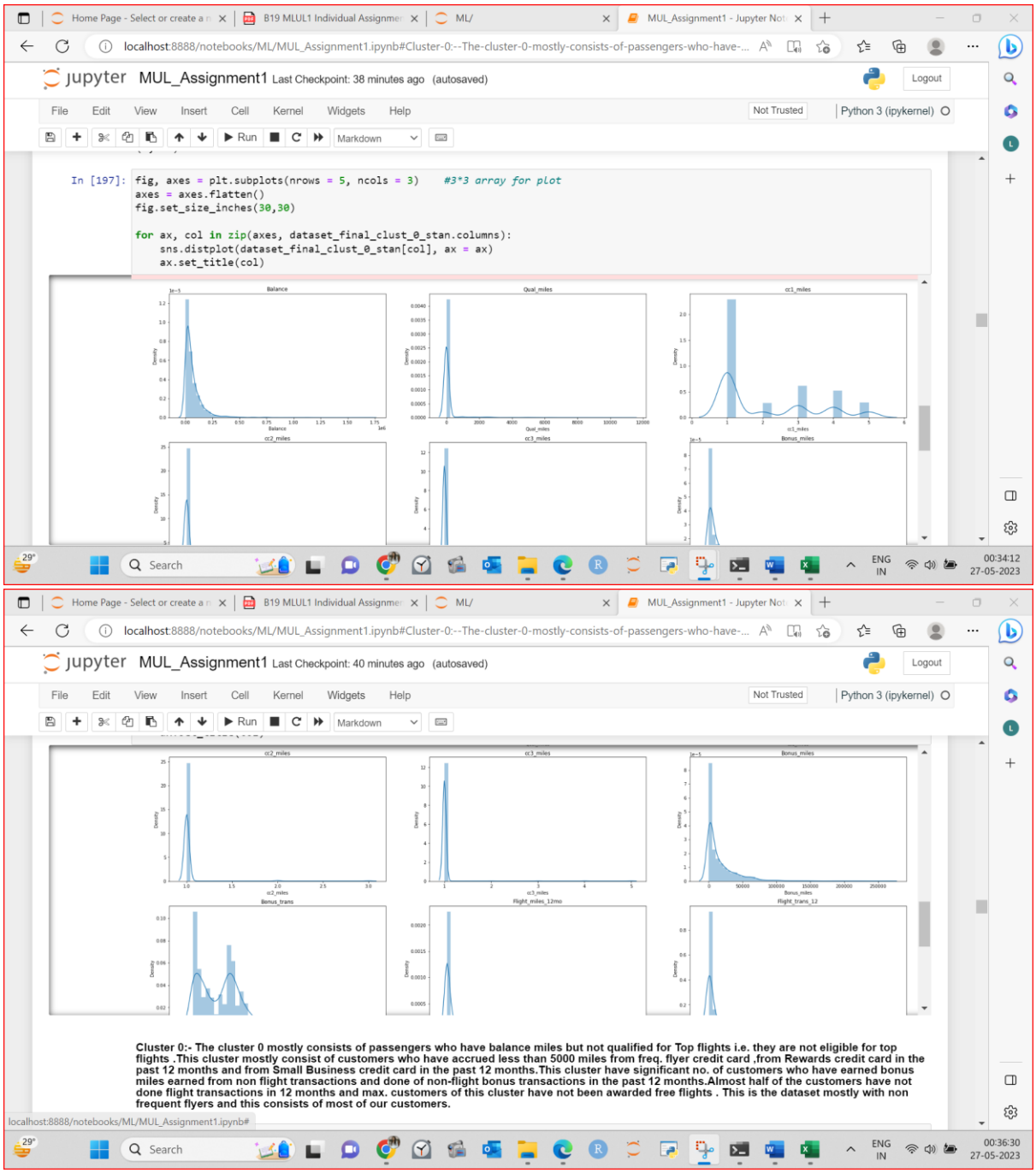
Creating Clusters using Standardization

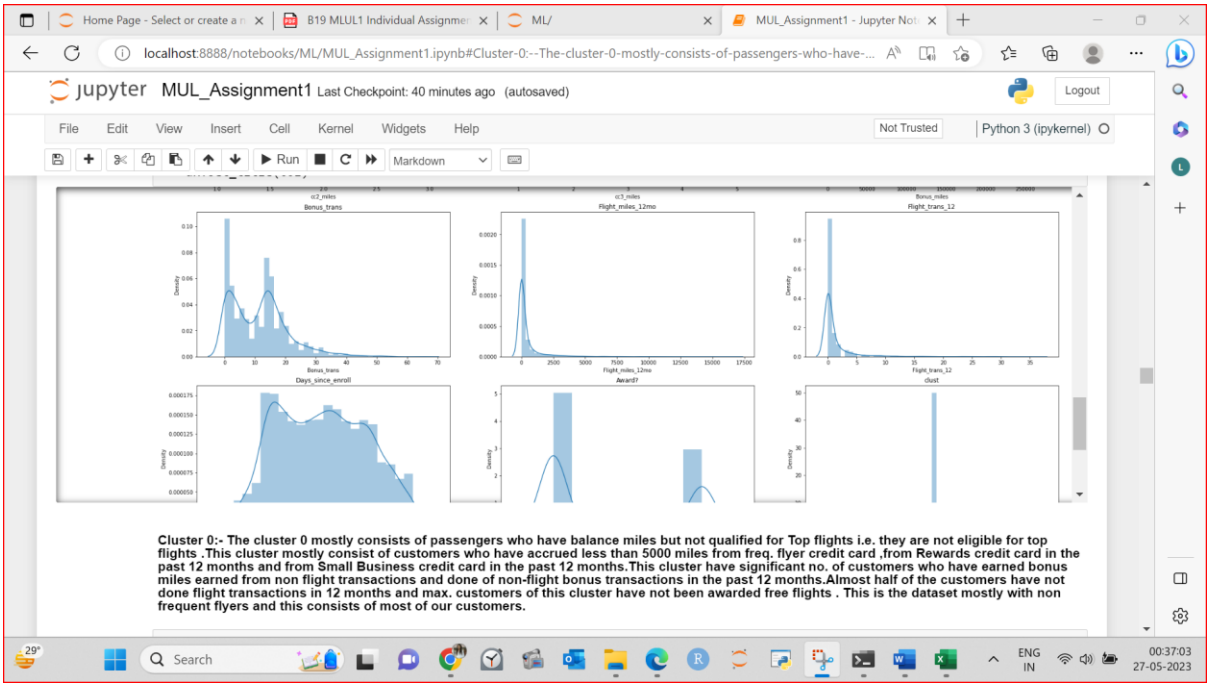
```
In [245]: from sklearn.cluster import AgglomerativeClustering
          h_complete = AgglomerativeClustering(n_clusters=2,linkage='complete',affinity = "euclidean").fit(dataset_scaled_stan)
          h_complete.fit_predict(dataset_scaled_stan)

Out[245]: array([0, 0, 0, ..., 0, 0, 0], dtype=int64)

In [246]: cluster_labels=pd.Series(h_complete.labels_)
          dataset_scaled_stan['clust']=cluster_labels # creating a  new column and assigning it to new column
          print(dataset_scaled_stan)
```

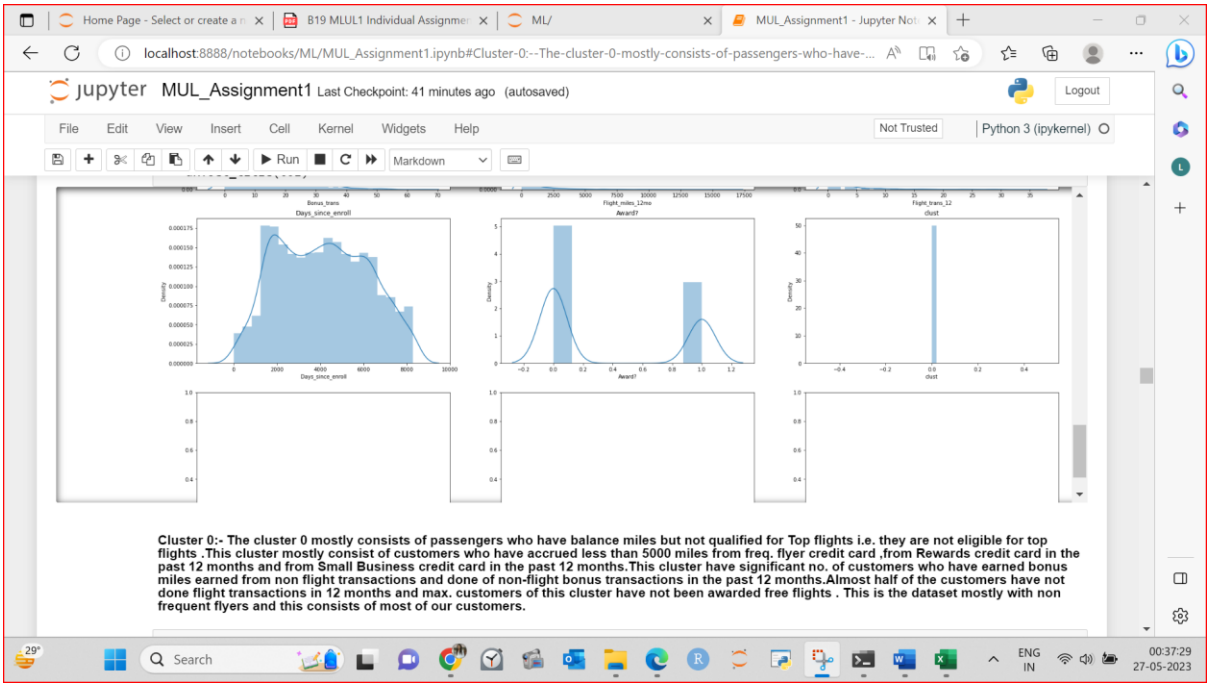
	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans
0	-0.186299	-0.769578	-0.098242	-0.062767	-0.702786	-1.104065
1	-0.186299	-0.769578	-0.098242	-0.062767	-0.701088	-0.999926
2	-0.186299	-0.769578	-0.098242	-0.062767	-0.539253	-0.791649
3	-0.186299	-0.769578	-0.098242	-0.062767	-0.689286	-1.104065
4	-0.186299	1.409471	-0.098242	-0.062767	1.083121	1.499394
...	...	...	...	...	...	...
3994	-0.186299	-0.769578	-0.098242	-0.062767	-0.356960	-0.791649
3995	-0.186299	-0.769578	-0.098242	-0.062767	-0.669367	-0.687511
3996	-0.186299	0.683121	-0.098242	-0.062767	0.343804	-0.375096
3997	-0.186299	-0.769578	-0.098242	-0.062767	-0.689286	-1.104065
3998	-0.186299	-0.769578	-0.098242	-0.062767	-0.709992	-1.208203

	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?	clust

Cluster 0:

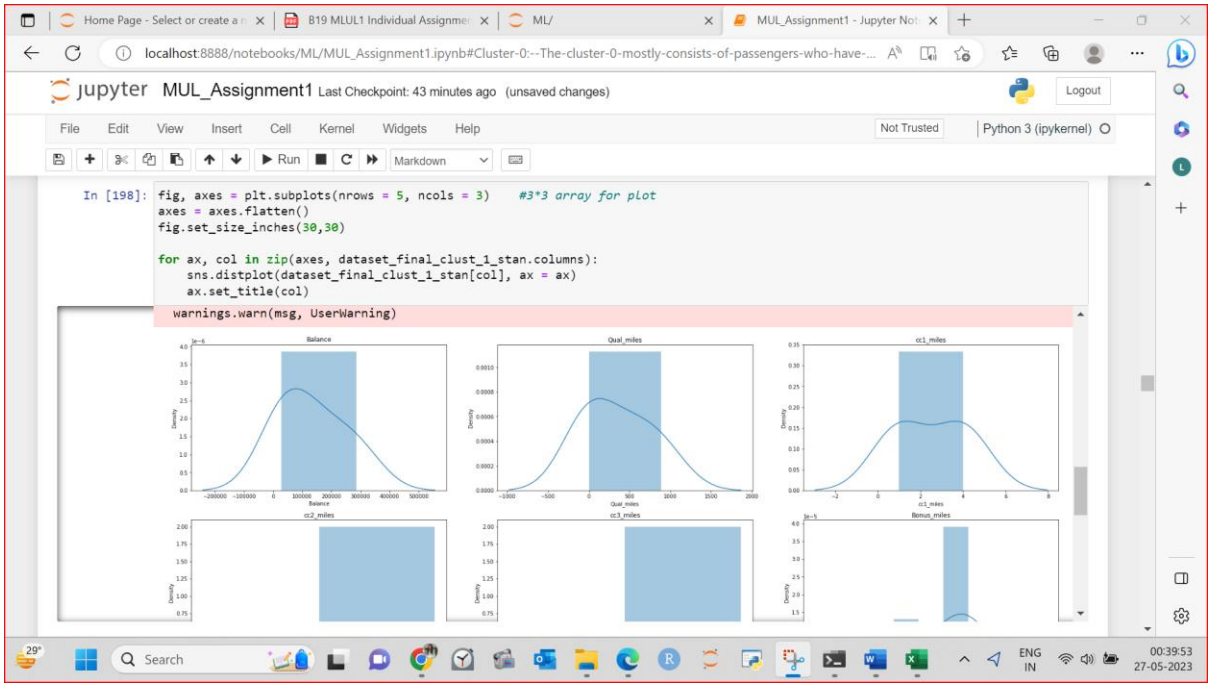

MUL_Assignment1 Last Checkpoint: 38 minutes ago (autosaved)
In [197]:
fig, axes = plt.subplots(nrows = 5, ncols = 3) #3*3 array for plot
axes = axes.flatten()
fig.set_size_inches(30,30)
for ax, col in zip(axes, dataset_final_clust_0_stan.columns):
sns.distplot(dataset_final_clust_0_stan[col], ax = ax)
ax.set_title(col)
Balance
Qual_miles
cc1_miles
cc2_miles
cc3_miles
Bonus_miles
MUL_Assignment1 Last Checkpoint: 40 minutes ago (autosaved)
Bonus_trans
Flight_miles_12mo
Flight_trans_12
Cluster 0:- The cluster 0 mostly consists of passengers who have balance miles but not qualified for Top flights i.e. they are not eligible for top flights .This cluster mostly consist of customers who have accrued less than 5000 miles from freq. flyer credit card ,from Rewards credit card in the past 12 months and from Small Business credit card in the past 12 months.This cluster have significant no. of customers who have earned bonus miles earned from non flight transactions and done of non-flight bonus transactions in the past 12 months.Almost half of the customers have not done flight transactions in 12 months and max. customers of this cluster have not been awarded free flights . This is the dataset mostly with non frequent flyers and this consists of most of our customers.

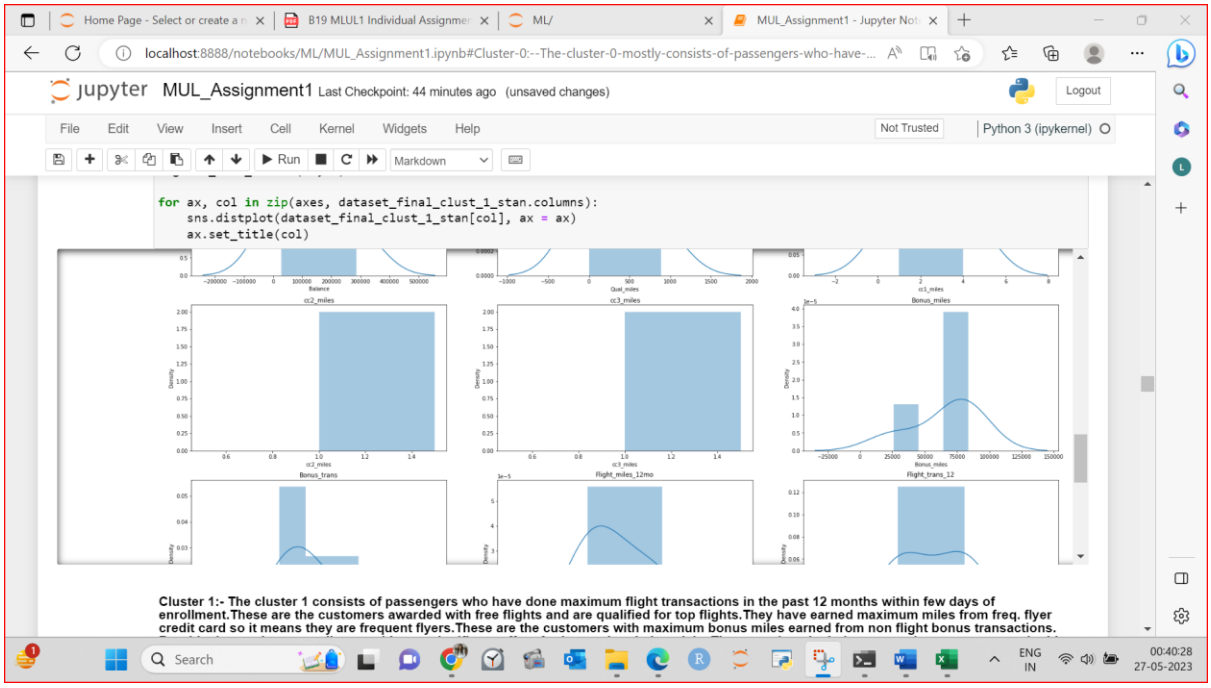


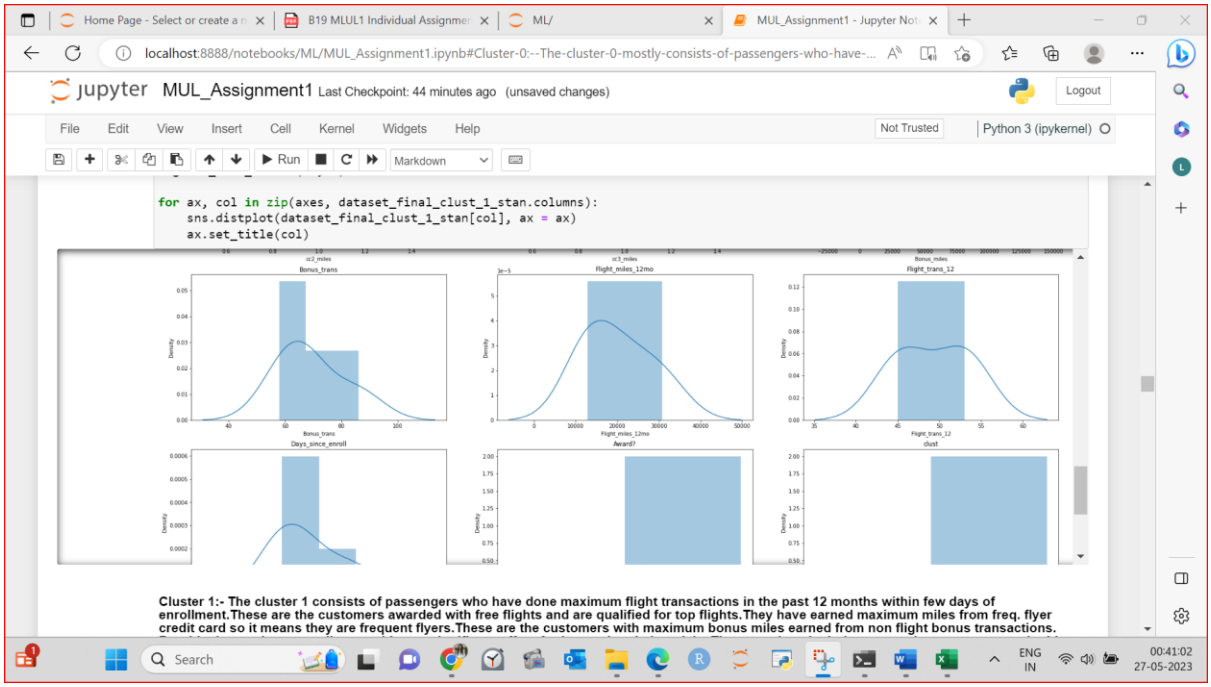


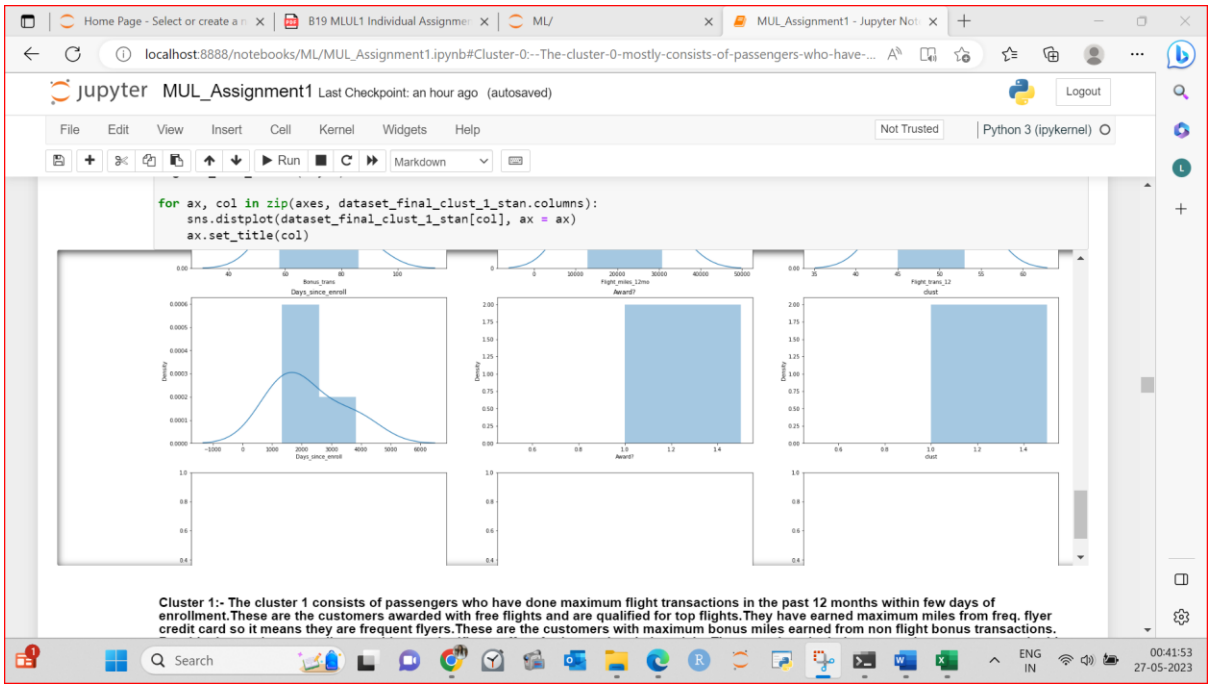
1. Cluster 0: - The cluster 0 mostly consists of passengers who have balance miles but not qualified for Top flights i.e. they are not eligible for top flights .This cluster mostly consist of customers who have accrued less than 5000 miles from freq. flyer credit card ,from Rewards credit card in the past 12 months and from Small Business credit card in the past 12 months. This cluster have significant no. of customers who have earned bonus miles earned from non-flight transactions and done of non-flight bonus transactions in the past 12 months. Almost half of the customers have not done flight transactions in 12 months and max. customers of this cluster have not been awarded free flights. This is the dataset mostly with non-frequent flyers and this consists of most of our customers.

Cluster 1:


MUL_Assignment1 Last Checkpoint: 43 minutes ago (unsaved changes)
In [198]:
fig, axes = plt.subplots(nrows = 5, ncols = 3) #3*3 array for plot
axes = axes.flatten()
fig.set_size_inches(30,30)
for ax, col in zip(axes, dataset_final_clust_1_stan.columns):
sns.distplot(dataset_final_clust_1_stan[col], ax = ax)
ax.set_title(col)
warnings.warn(msg, UserWarning)
Balance
Qual_miles
cc1_miles
cc2_miles
cc3_miles
Bonus_miles


MUL_Assignment1 Last Checkpoint: 44 minutes ago (unsaved changes)
for ax, col in zip(axes, dataset_final_clust_1_stan.columns):
sns.distplot(dataset_final_clust_1_stan[col], ax = ax)
ax.set_title(col)
Bonus_trans
Flight_miles_12mo
Flight_trans_12
Cluster 1:- The cluster 1 consists of passengers who have done maximum flight transactions in the past 12 months within few days of enrollment.These are the customers awarded with free flights and are qualified for top flights.They have earned maximum miles from freq. flyer credit card so it means they are frequent flyers.These are the customers with maximum bonus miles earned from non flight bonus transactions.



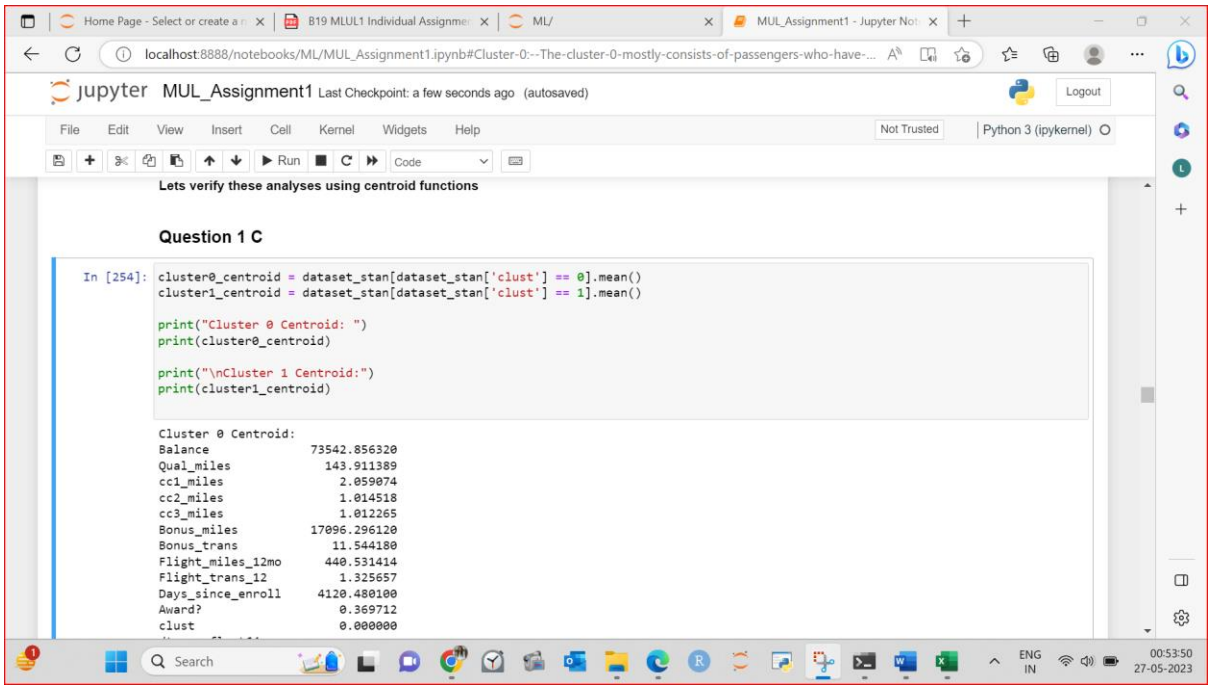


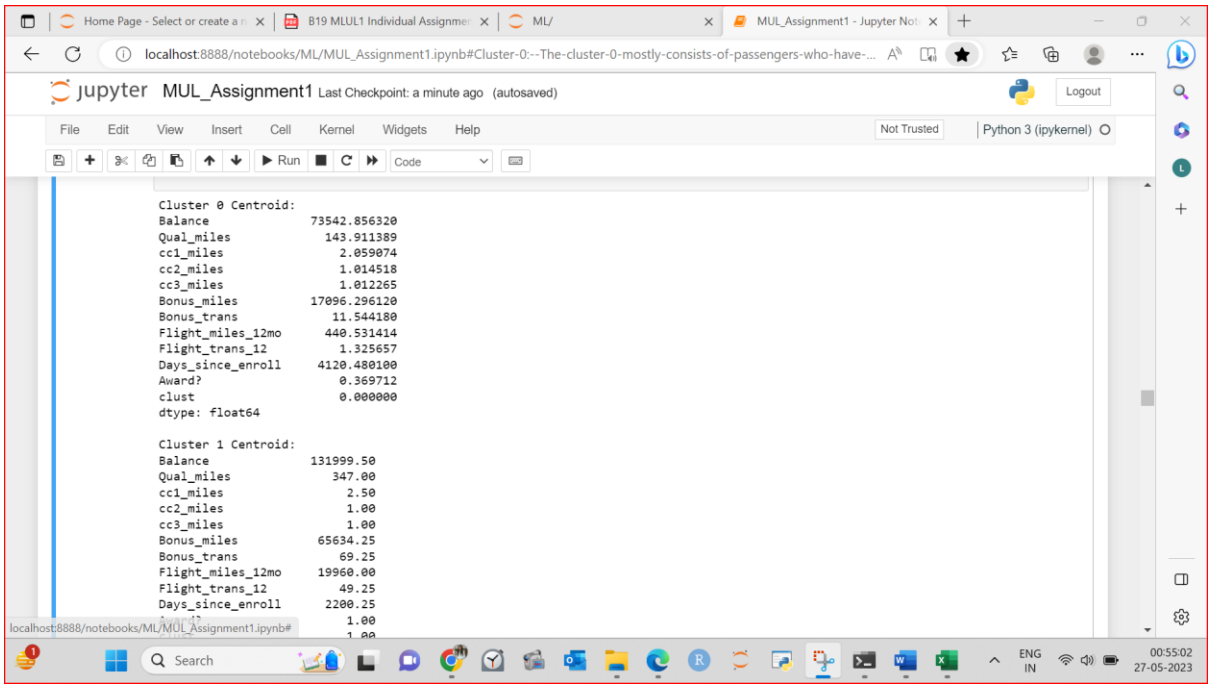
2. Cluster 1: - The cluster 1 consists of passengers who have done maximum flight transactions in the past 12 months within few days of enrolment. These are the customers awarded with free flights and are qualified for top flights. They have earned maximum miles from freq. flyer credit card, so it means they are frequent flyers. These are the customers with maximum bonus miles earned from non-flight bonus transactions. But this dataset is too small to provide any significant offers for increasing their activity. The target is to include more and more customers in this category as there are customers with higher balance miles but with less flight transactions. So these can be our target customers to provide offers to encourage them to be part of our frequent flyers customer.

This conclusion of mine shall be verified in the next question wherein centroid clusters are used to characterize both the clusters.

Question C:

Following code results in centroid of both the clusters and analysis of the same has been mentioned below:



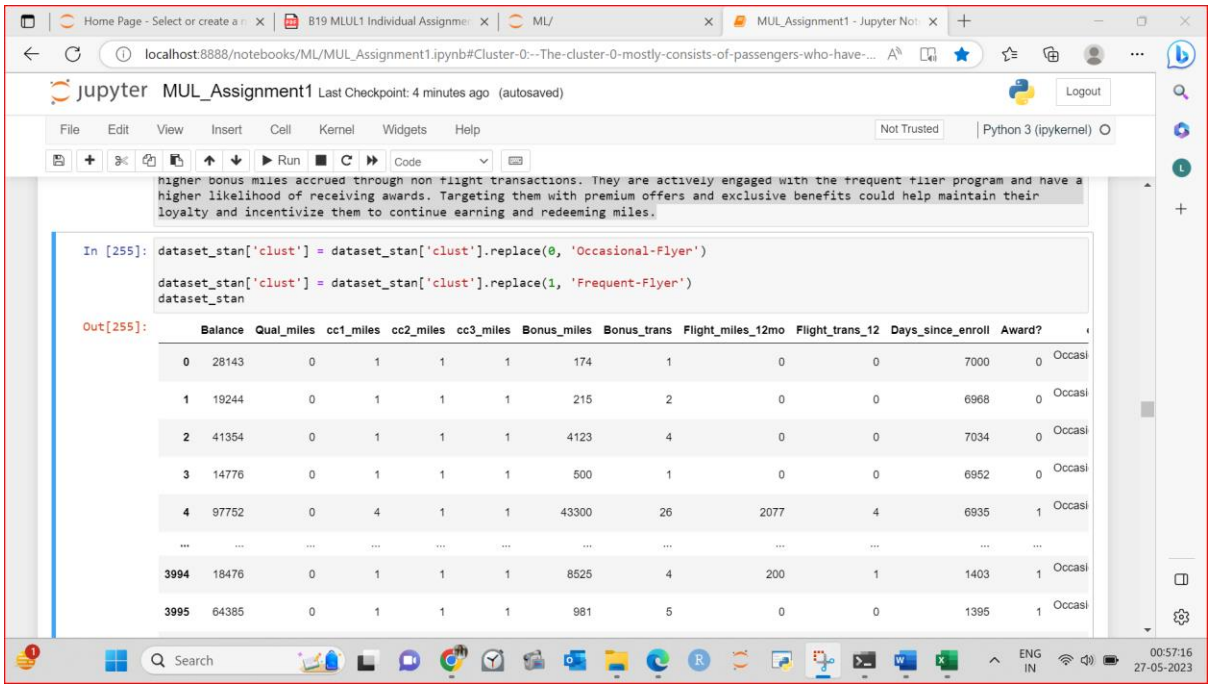


Cluster 0:

Occasional Flyers: This cluster represents passengers who have relatively lower balance miles, fewer miles to qualify for top flights, and lower bonus miles accrued through non flight transactions. These customers may not be highly engaged with the frequent flier program and have a lower likelihood of receiving awards. Targeting them with offers that encourage more flight and credit card usage could help increase their activity.

Cluster 1:

Frequent Flyers: This cluster represents passengers who have higher balance miles , more miles to qualify for top flights, and higher bonus miles accrued through non flight transactions. They are actively engaged with the frequent flier program and have a higher likelihood of receiving awards. Targeting them with premium offers and exclusive benefits could help maintain their loyalty and incentivize them to continue earning and redeeming miles.

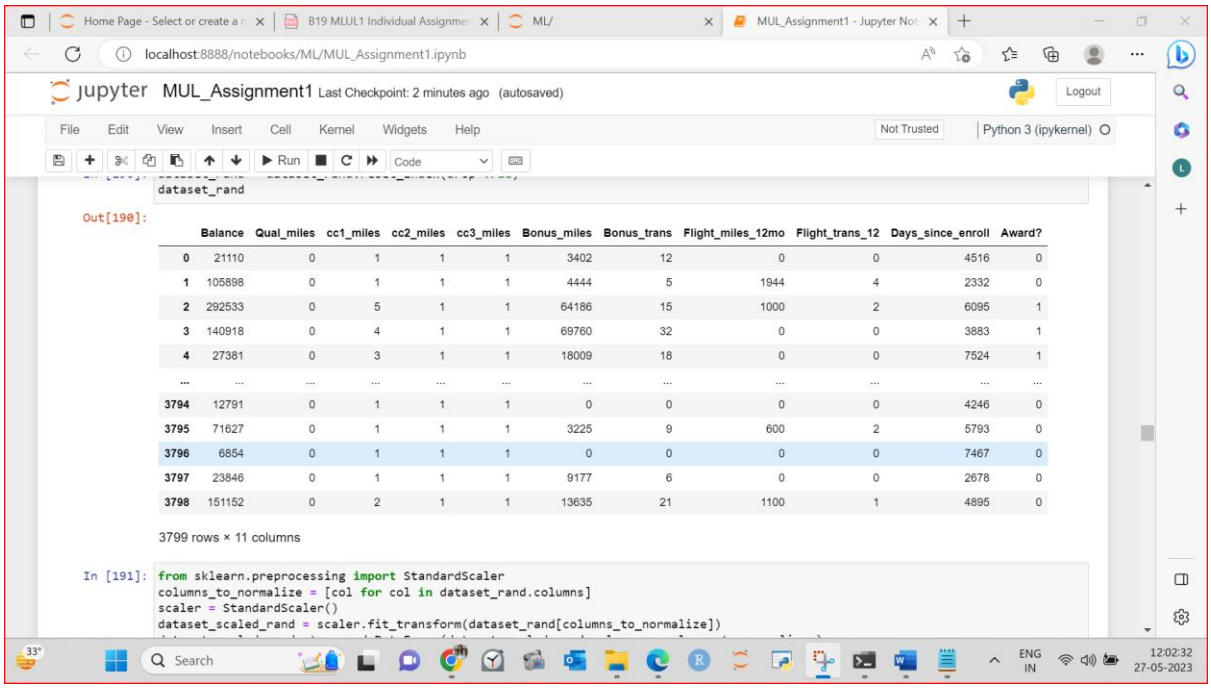


The above output is saved in **Snigdha_Bhattacharjee_12220067_1c.csv** files.

Question D:

Analysing the effect of sampling on clustering:

- Taken the 95% of the 4000 data



```
dataset_rand
```

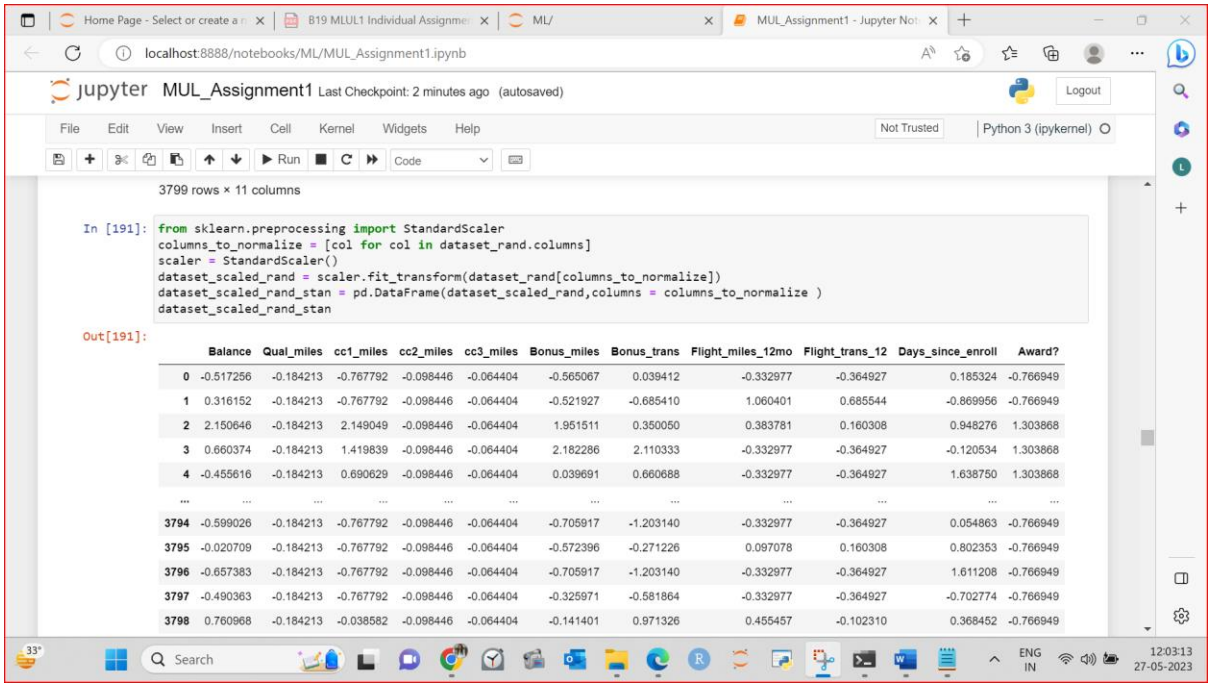
Out[190]:

	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?
0	21110	0	1	1	1	3402	12	0	0	4516	0
1	105898	0	1	1	1	4444	5	1944	4	2332	0
2	292533	0	5	1	1	64186	15	1000	2	6095	1
3	140918	0	4	1	1	69760	32	0	0	3883	1
4	27381	0	3	1	1	18009	18	0	0	7524	1
...	...	...	...	...	...	...	...	...	...	...	...
3794	12791	0	1	1	1	0	0	0	0	4246	0
3795	71627	0	1	1	1	3225	9	600	2	5793	0
3796	6854	0	1	1	1	0	0	0	0	7467	0
3797	23846	0	1	1	1	9177	6	0	0	2678	0
3798	151152	0	2	1	1	13635	21	1100	1	4895	0

3799 rows × 11 columns

In [191]:
```
from sklearn.preprocessing import StandardScaler
columns_to_normalize = [col for col in dataset_rand.columns]
scaler = StandardScaler()
dataset_scaled_rand = scaler.fit_transform(dataset_rand[columns_to_normalize])
```

- Applying Standardization techniques: StandardScaler.



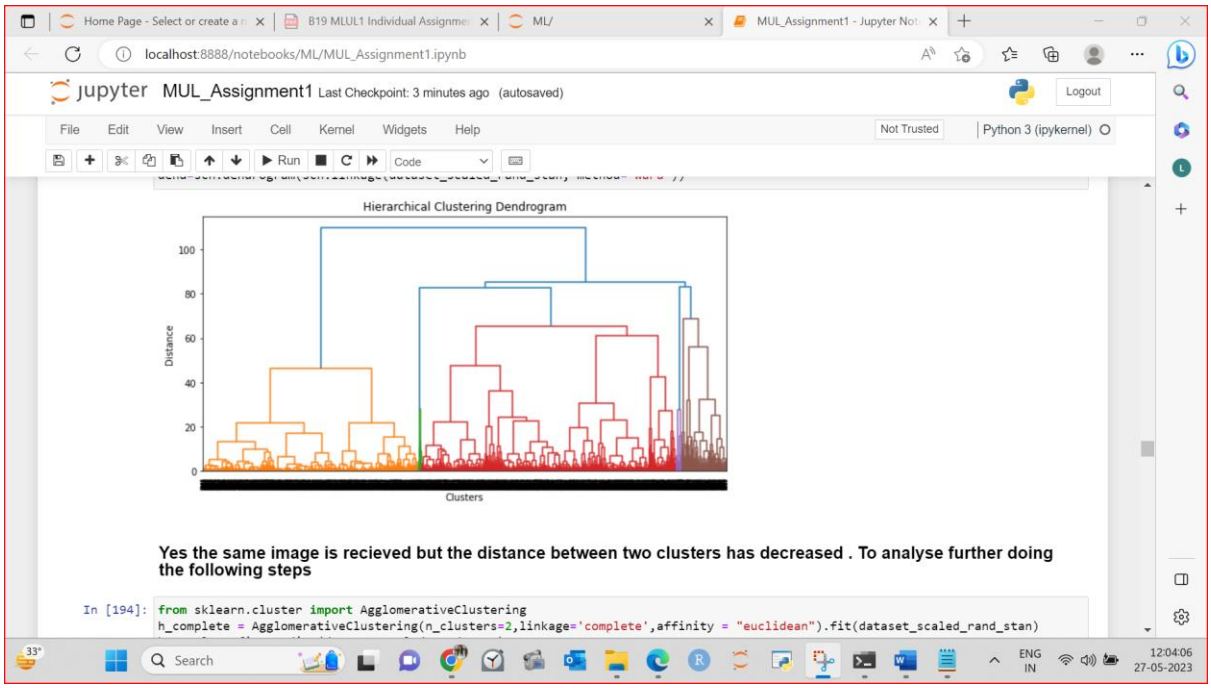
3799 rows × 11 columns

In [191]:
```
from sklearn.preprocessing import StandardScaler
columns_to_normalize = [col for col in dataset_rand.columns]
scaler = StandardScaler()
dataset_scaled_rand = scaler.fit_transform(dataset_rand[columns_to_normalize])
dataset_scaled_rand_stan = pd.DataFrame(dataset_scaled_rand,columns = columns_to_normalize )
dataset_scaled_rand_stan
```

Out[191]:

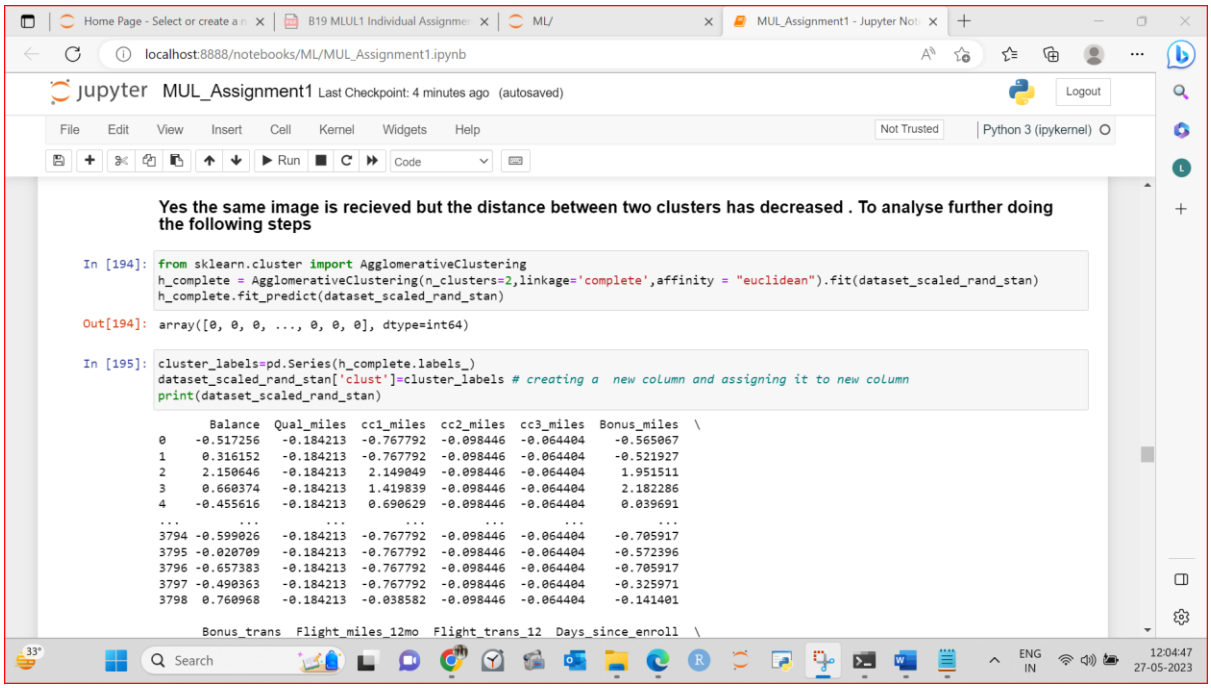
	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?
0	-0.517256	-0.184213	-0.767792	-0.098446	-0.064404	-0.565067	0.039412	-0.332977	-0.364927	0.185324	-0.766949
1	0.316152	-0.184213	-0.767792	-0.098446	-0.064404	-0.521927	-0.685410	1.060401	0.685544	-0.869956	-0.766949
2	2.150646	-0.184213	2.149049	-0.098446	-0.064404	1.951511	0.350050	0.383781	0.160308	0.948276	1.303868
3	0.660374	-0.184213	1.419839	-0.098446	-0.064404	2.182286	2.110333	-0.332977	-0.364927	-0.120534	1.303868
4	-0.455616	-0.184213	0.690629	-0.098446	-0.064404	0.039691	0.660688	-0.332977	-0.364927	1.638750	1.303868
...	...	...	...	...	...	...	...	...	...	...	...
3794	-0.599026	-0.184213	-0.767792	-0.098446	-0.064404	-0.705917	-1.203140	-0.332977	-0.364927	0.054863	-0.766949
3795	-0.020709	-0.184213	-0.767792	-0.098446	-0.064404	-0.572396	-0.271226	0.097078	0.160308	0.802353	-0.766949
3796	-0.657383	-0.184213	-0.767792	-0.098446	-0.064404	-0.705917	-1.203140	-0.332977	-0.364927	1.611208	-0.766949
3797	-0.490363	-0.184213	-0.767792	-0.098446	-0.064404	-0.325971	-0.581864	-0.332977	-0.364927	-0.702774	-0.766949
3798	0.760968	-0.184213	-0.038582	-0.098446	-0.064404	-0.141401	0.971326	0.455457	-0.102310	0.368452	-0.766949

- Applying hierarchical clustering with Euclidean distance and Ward's method to check the number of clusters we are getting.

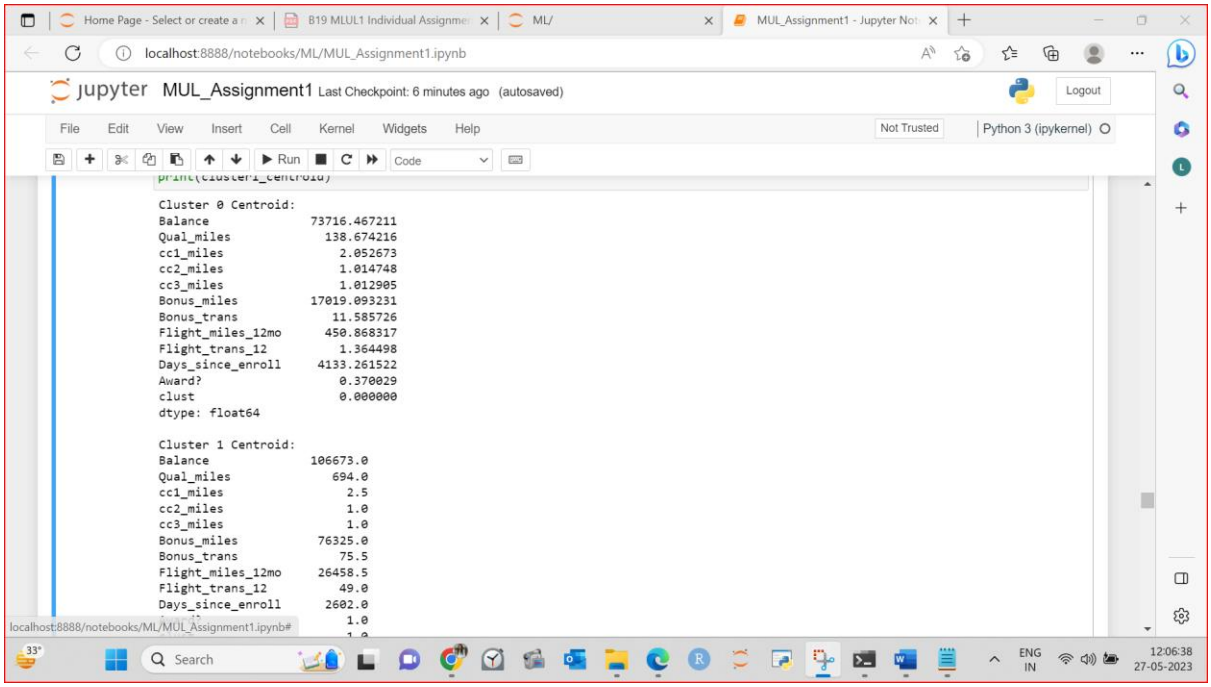


The output represents the same 2 major clusters but this time the distance between the two clusters has reduced i.e. the effect of sampling.

- Applying Agglomerative clustering for 2 clusters and assigning the cluster label to the datasets.



- Description and label of both these clusters are obtained using centroid clustering:



The centroid data has changed slightly but overall view of each of the cluster remained the same.

Cluster 0:

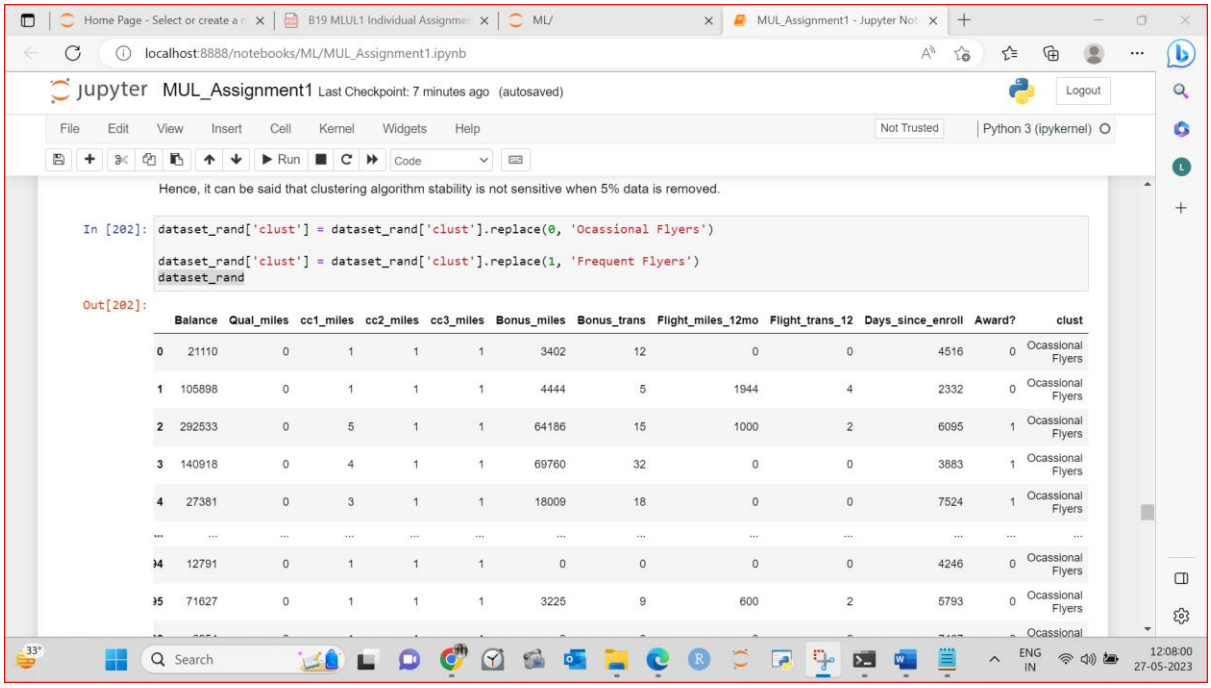
Ocassional Flyers: These customers have a moderate balance of eligible miles for award travel and engage in occasional flying miles qualify for top flights. They have a moderate activity level with credit cards and earn a moderate amount of bonus miles. They also have a moderate number of flight miles and transactions in the past 12 months. These customers have been enrolled for a relatively long period, and their likelihood of receiving an award flight is relatively low.

Cluster 1:

Frequent Flyers: These customers have a relatively high balance of eligible miles for award travel and have a large number of qualifying miles. They have a high activity level with credit cards earning them miles between 5000-25000 miles. They earn a significant amount of bonus miles through non flight transactions and have a high number of non flight transactions. Morever, they have high number flight miles or transactions in the past 12 months. These customers have been enrolled for a relatively shorter period, and have received an awarded flight.Considering these characteristics, we can label this cluster as Frequent Active Flyers.

Hence, it can be said that clustering algorithm stability is not sensitive when 5% data is removed.

- Applying labels into dataset based on the above analysis report:

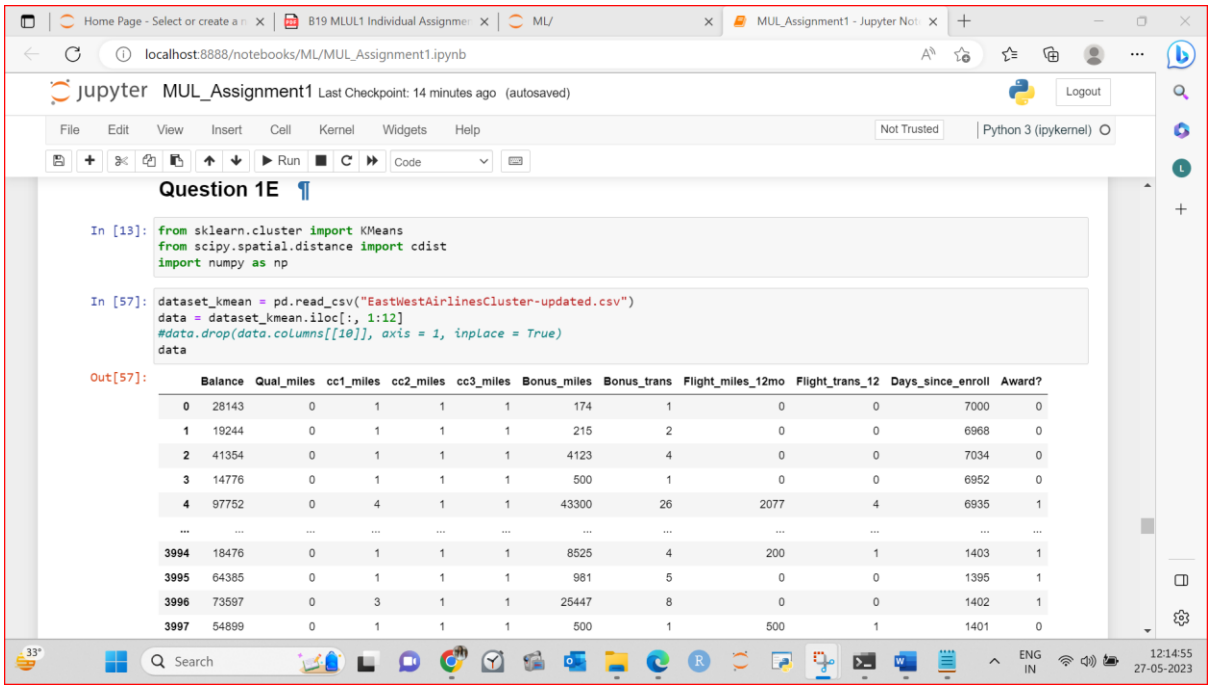


	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?	clust
0	21110	0	1	1	1	3402	12	0	0	4516	0	Ocassional Flyers
1	105898	0	1	1	1	4444	5	1944	4	2332	0	Ocassional Flyers
2	292533	0	5	1	1	64186	15	1000	2	6095	1	Ocassional Flyers
3	140918	0	4	1	1	69760	32	0	0	3883	1	Ocassional Flyers
4	27381	0	3	1	1	18009	18	0	0	7524	1	Ocassional Flyers
...	...	...	...	...	...	...	...	...	...	...	...	...
3994	12791	0	1	1	1	0	0	0	0	4246	0	Ocassional Flyers
3995	71627	0	1	1	1	3225	9	600	2	5793	0	Ocassional Flyers

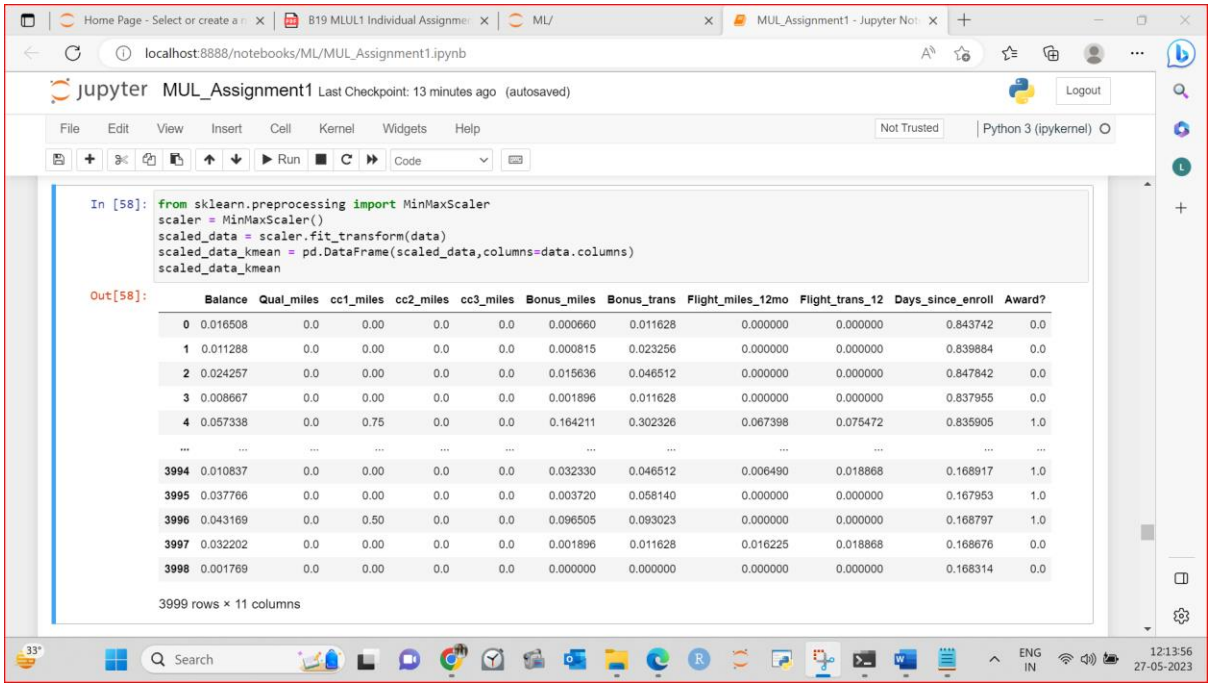
The above data is saved in the **Snigdha_Bhattacharjee_12220067_1d.csv** file

Question E:

Clustering all the passengers of the dataset using K-Means. Standardization applied on the cluster is MinMax Cluster.



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



```
In [58]: from sklearn.preprocessing import MinMaxScaler
         scaler = MinMaxScaler()
         scaled_data = scaler.fit_transform(data)
         scaled_data_kmean = pd.DataFrame(scaled_data,columns=data.columns)
         scaled_data_kmean
```

Out[58]:

	Balance	Qual_miles	cc1_miles	cc2_miles	cc3_miles	Bonus_miles	Bonus_trans	Flight_miles_12mo	Flight_trans_12	Days_since_enroll	Award?
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

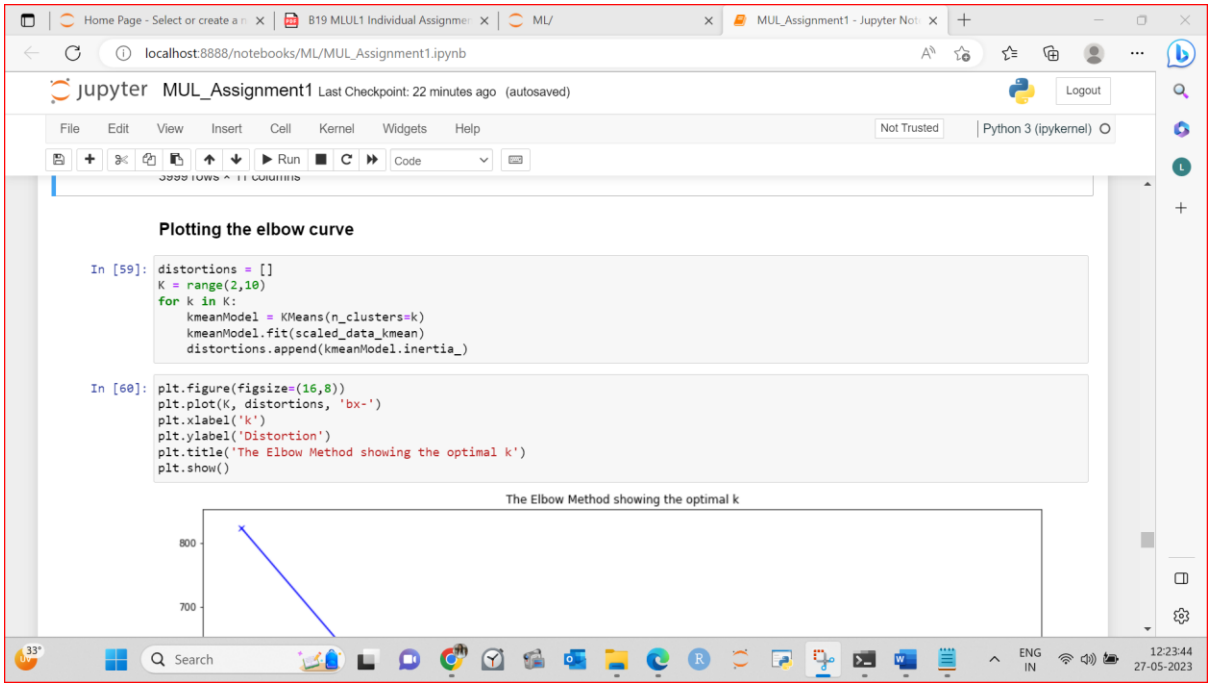
Now applying K-mean Clustering and to do so the number of clusters is decided using elbow method.

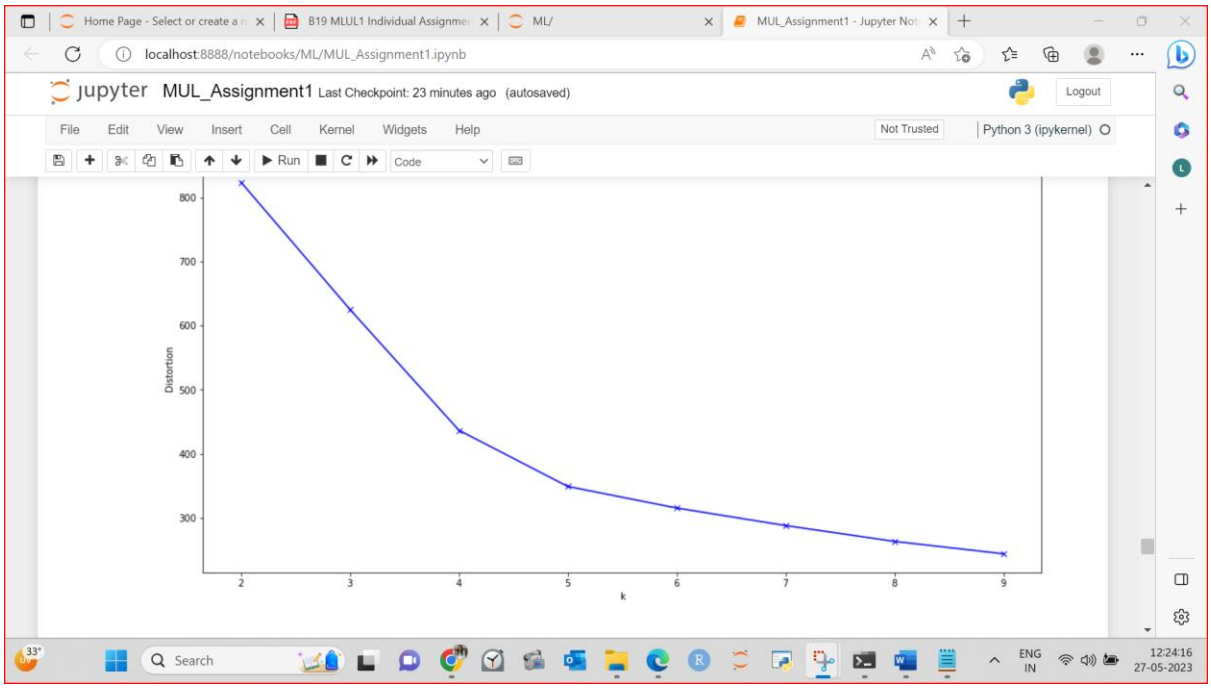
The elbow method or elbow curve is a graphical technique to determine the optimal number of clusters in a K-means clustering algorithm.

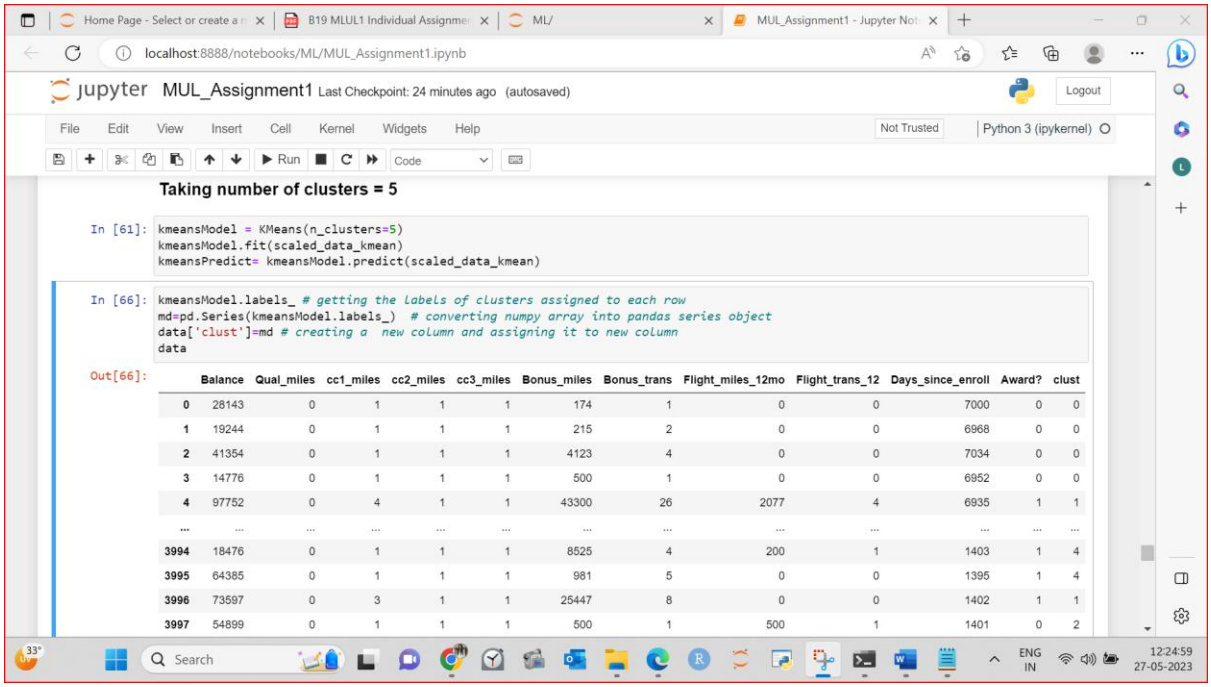
The elbow curve represents the relationship between the number of clusters (K) on the x-axis and the distortion (or inertia) on the y-axis.

The shape of the elbow curve resembles an arm bent at the elbow. As the number of clusters increases, the distortion typically decreases because each data point can be assigned to a closer centroid. However, at some point, adding more clusters has diminishing returns, and the rate of decrease in distortion slows down this is where the elbow of the curve occurs.

By analysing the elbow curve of my dataset, the optimal number of clusters is 5.


Home Page - Select or create a
B19 MLUL1 Individual Assignme
ML/
MUL_Assignment1 - Jupyter Not
localhost:8888/notebooks/ML/MUL_Assignment1.ipynb
jupyter MUL_Assignment1 Last Checkpoint: 22 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Plotting the elbow curve
In [59]:
distortions = []
K = range(2,10)
for k in K:
kmeanModel = KMeans(n_clusters=k)
kmeanModel.fit(scaled_data_kmean)
distortions.append(kmeanModel.inertia_)
In [60]:
plt.figure(figsize=(16,8))
plt.plot(K, distortions, 'bx-')
plt.xlabel('k')
plt.ylabel('Distortion')
plt.title('The Elbow Method showing the optimal k')
plt.show()
The Elbow Method showing the optimal k
800
700
Search
ENG
IN
12:23:44
27-05-2023

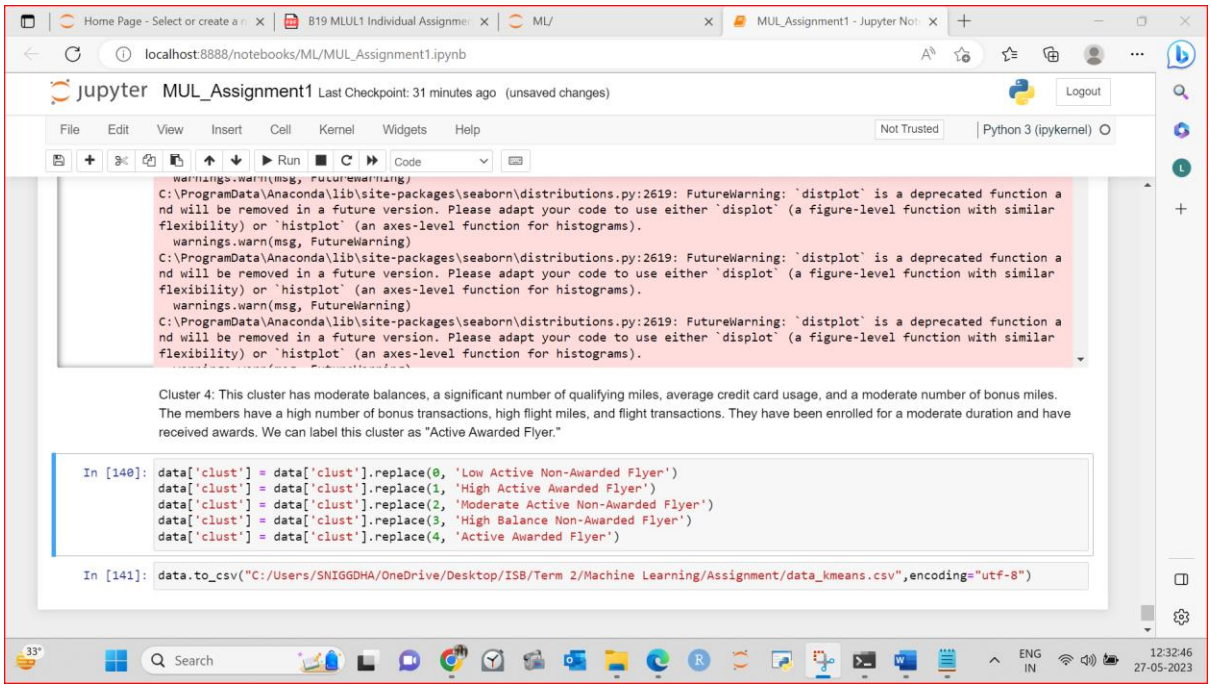

Home Page - Select or create a
B19 MLUL1 Individual Assignme
ML/
MUL_Assignment1 - Jupyter Not
localhost:8888/notebooks/ML/MUL_Assignment1.ipynb
jupyter MUL_Assignment1 Last Checkpoint: 23 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
800
700
600
500
400
300
Distortion
2
3
4
5
6
7
8
9
k
Search
ENG
IN
12:24:16
27-05-2023



In the above picture I have assigned cluster labels as 0,1,2,3,4. Now analysing each cluster to provide appropriate cluster names.

- Cluster 0: This cluster has relatively lower balances, less qualifying miles to qualify for top flights, low usage of credit cards (cc1_miles, cc2_miles, cc3_miles), and fewer bonus miles. The members of this cluster have a low number of bonus transactions, flight miles, and flight transactions. They have been enrolled for a long time but have not received any awards. We will label this cluster as "**Low Activity Non-Award Members**."
- Cluster 1: This cluster stands out with high balances, a significant number of customers with qualifying miles, high credit card usage, and a large number of bonus miles. The members of this cluster have high bonus transactions, flight miles, and flight transactions. They have been enrolled for a considerable period and have received awards. We can label this cluster as "**High Activity Award-Winning Members.**"
- Cluster 2: This cluster shows moderate balances, a moderate number of qualifying miles, and average credit card usage. The members have a moderate number of bonus miles, bonus transactions, flight miles, and flight transactions. They have been enrolled for a moderate duration but have not received any awards. We can label this cluster as "**Moderate Activity Non-Award Members**."
- Cluster 3: This cluster has relatively high balances, a small number of qualifying miles, and high credit card usage. The members have a moderate number of bonus miles, high bonus transactions, low flight miles, and a small number of flight transactions. They have been enrolled for a considerable period but have not received any awards. We can label this cluster as "**High Balance Non-Award Members.**"
- Cluster 4: This cluster has moderate balances, a significant number of qualifying miles, average credit card usage, and a moderate number of bonus miles. The members have a high number of bonus transactions, high flight miles, and flight transactions. They have been enrolled for a moderate duration and have received awards. We can label this cluster as "**Active Awarded Flyer**."

The graphical representation of above 5 clusters is in the python notebook file. Assigning the above clusters derived name into the dataset:



Refer output excel file **Snigdha_Bhattacharjee_12220067_1e.csv file.**

Question F:

Comparing the clusters obtained using K-Means and Hierarchical Clustering:

- Cluster 0 in both K-means and hierarchical clustering: In both cases, Cluster 0 represents customers with relatively lower balances, lower qualifying miles, and low engagement with credit cards and bonus transactions. They have been enrolled for a long time but have not received any awards. This cluster is labelled as "Low Activity Non-Award Members" in K-means clustering and "Occasional Flyers" in hierarchical clustering. Both algorithms identify this segment as less active and less likely to receive awards.
- Cluster 1 in both K-means and hierarchical clustering: In both cases, Cluster 1 represents customers with high balances, a significant number of qualifying miles, high credit card usage, and many bonus miles. They have been enrolled for a considerable period and have received awards. This cluster is labelled as "High Activity Award-Winning Members" in K-means clustering and "Frequent Flyers" in hierarchical clustering. Both algorithms identify this segment as highly active and likely to receive awards.
- Cluster 2 in K-means clustering: K-means clustering identifies Cluster 2 as having moderate balances, a moderate number of qualifying miles, and average credit card usage. The members have a moderate number of bonus miles, bonus transactions, flight miles, and flight transactions. They have been enrolled for a moderate duration but have not received any awards. This cluster is labelled as "Moderate Activity Non-Award Members" in K-means clustering.

- Cluster 3 and Cluster 4 in K-means clustering: K-means clustering identifies Cluster 3 as "High Balance Non-Award Members" and Cluster 4 as "Active Awarded Flyers." These clusters have different characteristics in terms of qualifying miles, bonus transactions, flight miles, and flight transactions. However, hierarchical clustering does not explicitly identify similar clusters with these characteristics.

In K- Mean Clustering the dataset was not divided into just two categories Occasional and Frequent flyers. It has divided passengers using combinations of Balance miles, spending nature in terms of miles and flight transactions and whether they are awarded free flights or not. This segmentation provides much more clear classification of passengers and provide Airlines much deeper understanding of their customers behaviour, which will enable airline to structure their program more efficiently.

Question G:

Cluster 1 (Frequent Flyers / High Active Awarded Flyers): This cluster represents customers with high balances, a significant number of qualifying miles, high credit card usage, and many bonus miles. They have been actively engaged with the frequent flyer program and have a higher likelihood of receiving awards. Targeting these passengers with premium offers and exclusive benefits such as exclusive lounge access, priority boarding, or upgrades would be beneficial to maintain their loyalty and incentivize them to continue earning and redeeming miles.

Cluster 4 (Active Awarded Flyers): This cluster has moderate balances, a significant number of qualifying miles, average credit card usage, and a moderate number of bonus miles. The members of this cluster have a high number of bonus transactions, high flight miles, and flight transactions. They have been enrolled for a moderate duration and have received awards. These are the active passengers as they have lower balance miles and high transactions so targeting them with offers that can enhance their travel experiences and offers for additional rewards such as offering bonus miles for specific flight routes, discounted, or upgraded travel packages, or targeted promotions for accumulating additional miles could encourage their continued engagement and loyalty.

Cluster 3 (High Balance Non-Awarded Flyers): This comprises customers with relatively high balances, a small number of qualifying miles, high credit card usage, and a small number of flight transactions. Despite being enrolled for a considerable period, they have not received any awards. These customers have the potential to transition into one of the previously mentioned clusters if targeted with offers that encourage their activity. offering bundled deals that combine flights, accommodations, and other travel services can motivate them to plan and book trips, utilizing their high balances in the process. Targeted promotions highlighting the benefits of redeeming accumulated miles for award travel can showcase the value and flexibility of their high balances, further encouraging them to utilize their miles for flights.