Customer Clustering based on Relationship with Card Providers

0. Setup Google Drive Environment and Get Data

```
!pip install -U -q PyDrive

from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials

auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)
```

```
file = drive.CreateFile({'id':'1NWnmneol2ApYRvl8uBYtFsNNbSDVWiz6'}) # replace the id with id of file you want to access
file.GetContentFile('data.csv')
```

1. Data Overview

```
import pandas as pd
import numpy as np
df = pd.read_csv('data.csv')
df.head()
```

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Statu
0	768805383	Existing Customer	45	M	3	High School	Marrie
1	818770008	Existing Customer	49	F	5	Graduate	Singl
2	713982108	Existing Customer	51	M	3	Graduate	Marrie
3	769911858	Existing Customer	40	F	4	High School	Unknow
4	709106358	Existing Customer	40	M	3	Uneducated	Marrie

5 rows × 23 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10127 entries, 0 to 10126
Data columns (total 23 columns):
 #   Column                                                                                                                Non-Nul
---  ------                                                                                                                -------
 0   CLIENTNUM                                                                                                             10127 n
 1   Attrition_Flag                                                                                                        10127 n
 2   Customer_Age                                                                                                          10127 n
 3   Gender                                                                                                                10127 n
 4   Dependent_count                                                                                                       10127 n
 5   Education_Level                                                                                                       10127 n
 6   Marital_Status                                                                                                        10127 n
 7   Income_Category                                                                                                       10127 n
 8   Card_Category                                                                                                         10127 n
 9   Months_on_book                                                                                                        10127 n
 10  Total_Relationship_Count                                                                                              10127 n
 11  Months_Inactive_12_mon                                                                                                10127 n
 12  Contacts_Count_12_mon                                                                                                 10127 n
 13  Credit_Limit                                                                                                          10127 n
 14  Total_Revolving_Bal                                                                                                   10127 n
 15  Avg_Open_To_Buy                                                                                                       10127 n
 16  Total_Amt_Chng_Q4_Q1                                                                                                  10127 n
 17  Total_Trans_Amt                                                                                                       10127 n
 18  Total_Trans_Ct                                                                                                        10127 n
 19  Total_Ct_Chng_Q4_Q1                                                                                                   10127 n
 20  Avg_Utilization_Ratio                                                                                                 10127 n
 21  Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_1  10127 n
 22  Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_2  10127 n
```

```
dtypes: float64(7), int64(10), object(6)
memory usage: 1.8+ MB
```

```
df.nunique()
```

```
CLIENTNUM                                                                                                                            10127
Attrition_Flag                                                                                                                           2
Customer_Age                                                                                                                            45
Gender                                                                                                                                   2
Dependent_count                                                                                                                          6
Education_Level                                                                                                                          7
Marital_Status                                                                                                                           4
Income_Category                                                                                                                          6
Card_Category                                                                                                                            4
Months_on_book                                                                                                                          44
Total_Relationship_Count                                                                                                                 6
Months_Inactive_12_mon                                                                                                                   7
Contacts_Count_12_mon                                                                                                                    7
Credit_Limit                                                                                                                          6205
Total_Revolving_Bal                                                                                                                   1974
Avg_Open_To_Buy                                                                                                                       6813
Total_Amt_Chng_Q4_Q1                                                                                                                  1158
Total_Trans_Amt                                                                                                                       5033
Total_Trans_Ct                                                                                                                         126
Total_Ct_Chng_Q4_Q1                                                                                                                    830
Avg_Utilization_Ratio                                                                                                                  964
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_1    1704
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_2     640
dtype: int64
```

```
df.isnull().sum()
```

```
CLIENTNUM                                                                                                                            0
Attrition_Flag                                                                                                                       0
Customer_Age                                                                                                                         0
Gender                                                                                                                               0
Dependent_count                                                                                                                      0
Education_Level                                                                                                                      0
Marital_Status                                                                                                                       0
Income_Category                                                                                                                      0
Card_Category                                                                                                                        0
Months_on_book                                                                                                                       0
Total_Relationship_Count                                                                                                             0
Months_Inactive_12_mon                                                                                                               0
Contacts_Count_12_mon                                                                                                                0
Credit_Limit                                                                                                                         0
Total_Revolving_Bal                                                                                                                  0
Avg_Open_To_Buy                                                                                                                      0
Total_Amt_Chng_Q4_Q1                                                                                                                 0
Total_Trans_Amt                                                                                                                      0
Total_Trans_Ct                                                                                                                       0
Total_Ct_Chng_Q4_Q1                                                                                                                  0
Avg_Utilization_Ratio                                                                                                                0
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_1    0
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_2    0
dtype: int64
```

As shown above, this data set has 10127 rows and 22 columns from 3 categories:

1.Demographic Information

CLIENTNUM: Unique identifier for each customer.

Customer_Age: Age of customer.

Gender: Gender of customer.

Dependent_count: Number of dependents that customer has.

Education_Level: Education level of customer.

Marital_Status: Marital status of customer.

Income_Category: Income category of customer.

2.Relationship with Card Provider

Card_Category: Type of card held by customer.

Months_on_book: How long customer has been on the books.

Total_Relationship_Count: Total number of relationships customer has with the credit card provider.

Months_Inactive_12_mon: Number of months customer has been inactive in the last twelve months.

Contacts_Count_12_mon: Number of contacts customer has had in the last twelve months.

Credit_Limit: Credit limit of customer.

3.Spending Behavior

Total_Revolving_Bal: Total revolving balance of customer.

Avg_Open_To_Buy: Average open to buy ratio of customer.

Total_Amt_Chng_Q4_Q1: Total amount changed from quarter 4 to quarter 1.

Total_Trans_Amt: Total transaction amount.

Total_Trans_Ct: Total transaction count.

Total_Ct_Chng_Q4_Q1: Total count changed from quarter 4 to quarter 1.

Avg_Utilization_Ratio: Average utilization ratio of customer.

4.Information for Prediction

Attrition_Flag: Flag indicating whether or not the customer has churned out.

Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon:

Naive Bayes classifier for predicting whether or not someone will churn

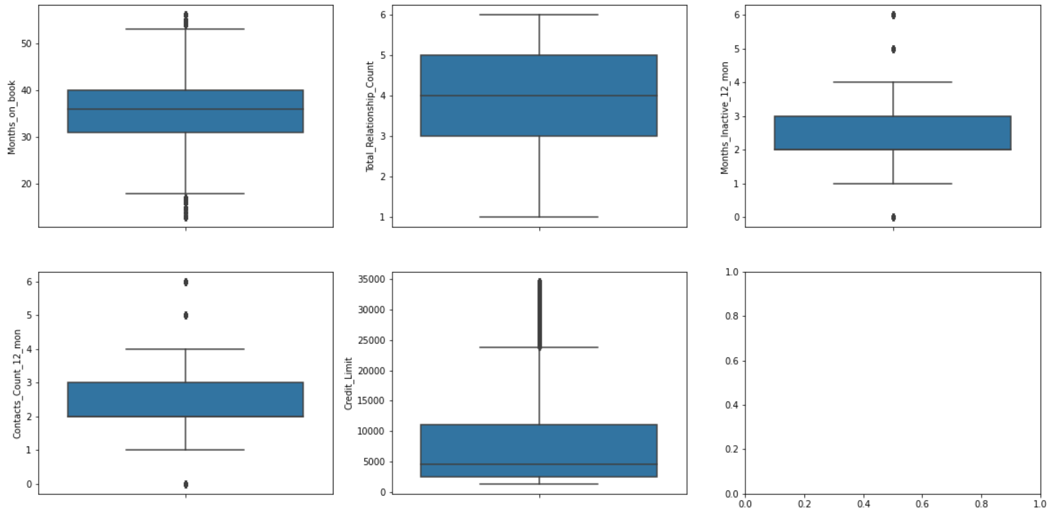
2. Relationship with Card Provider Preprocessing

```
df_cp = df[['Card_Category', 'Months_on_book', 'Total_Relationship_Count', 'Months_Inactive_12_mon', 'Contacts_Count_12_mon', 'Credit_Limit']]
df_cp.head()
```

	Card_Category	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon	Contacts_Count_
0	Blue	39	5	1	
1	Blue	44	6	1	
2	Blue	36	4	1	
3	Blue	34	3	4	
4	Blue	21	5	1	

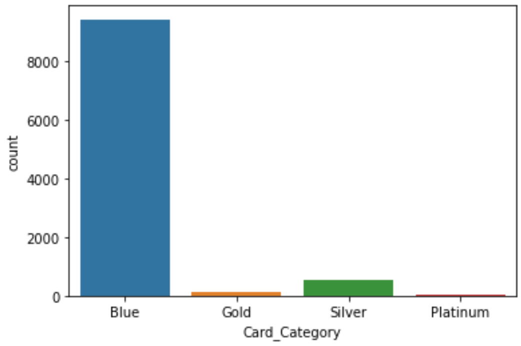
```
import matplotlib.pyplot as plt
import seaborn as sns
_,axss = plt.subplots(2,3, figsize=[20,10])
sns.boxplot(y ='Months_on_book', data=df_cp, ax=axss[0][0])
sns.boxplot(y ='Total_Relationship_Count', data=df_cp, ax=axss[0][1])
sns.boxplot(y ='Months_Inactive_12_mon', data=df_cp, ax=axss[0][2])
sns.boxplot(y ='Contacts_Count_12_mon', data=df_cp, ax=axss[1][0])
sns.boxplot(y ='Credit_Limit', data=df_cp, ax=axss[1][1])
```

<Axes: ylabel='Credit_Limit'>



```
sns.countplot(x = "Card_Category", data=df_cp)
```

```
<Axes: xlabel='Card_Category', ylabel='count'>
```



Although we have a lot of outliers here, we choose not to drop them for they can be important evidence for clustering.

Encoding Card Categories

```
num_cols = df_cp.columns[(df_cp.dtypes == 'float64') | (df_cp.dtypes == 'int64')]
cat_cols = df_cp.columns[df_cp.dtypes == 'object']
```

```
from sklearn.preprocessing import OrdinalEncoder

df_cp_Encoded = df_cp.copy()
categories = ['Card_Category']
enc_oe = OrdinalEncoder()
enc_oe.fit(df_cp_Encoded[categories])
df_cp_Encoded[categories] = enc_oe.transform(df_cp_Encoded[categories])
df_cp_Encoded.head()
```

	Card_Category	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon	Contacts_Count_
0	0.0	39	5	1	
1	0.0	44	6	1	
2	0.0	36	4	1	
3	0.0	34	3	4	
4	0.0	21	5	1	

```
df_cp.head()
```

	Card_Category	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon	Contacts_Count_
0	Blue	39	5	1	
1	Blue	44	6	1	
2	Blue	36	4	1	
3	Blue	34	3	4	
4	Blue	21	5	1	

```
df_cp_Encoded.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10127 entries, 0 to 10126
Data columns (total 6 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Card_Category             10127 non-null  float64
 1   Months_on_book            10127 non-null  int64
 2   Total_Relationship_Count  10127 non-null  int64
 3   Months_Inactive_12_mon    10127 non-null  int64
 4   Contacts_Count_12_mon     10127 non-null  int64
 5   Credit_Limit              10127 non-null  float64
dtypes: float64(2), int64(4)
memory usage: 474.8 KB
```

```
df_cp_Encoded.describe()
```

	Card_Category	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon	Contacts_Co
count	10127.000000	10127.000000	10127.000000	10127.000000	1(
mean	0.179816	35.928409	3.812580	2.341167	
std	0.693039	7.986416	1.554408	1.010622	
min	0.000000	13.000000	1.000000	0.000000	
25%	0.000000	31.000000	3.000000	2.000000	
50%	0.000000	36.000000	4.000000	2.000000	
75%	0.000000	40.000000	5.000000	3.000000	
max	3.000000	56.000000	6.000000	6.000000	

3. K-means Clustering Model

K-means Clustering Model with 5 Clusters

```
from sklearn.cluster import KMeans

num_clusters = 5 # this number is randomly choosed

# number of clusters
km = KMeans(n_clusters=num_clusters)
km.fit(df_cp_Encoded)

clusters = km.labels_.tolist()
clusters = pd.DataFrame(clusters)
```

```
/usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto
  warnings.warn(
```

```
df_cp['Cluster'] = clusters[0]
df_cp.head()
```

```
<ipython-input-9-9de0d0e88cbe>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#retur
  df_cp['Cluster'] = clusters[0]
```

	Card_Category	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon	Contacts_Count_
0	Blue	39	5	1	
1	Blue	44	6	1	
2	Blue	36	4	1	
3	Blue	34	3	4	
4	Blue	21	5	1	

```
print ("Number of reviews included in each cluster:")
df_cp['Cluster'].value_counts().to_frame()
```

Number of reviews included in each cluster:

	Cluster
2	5428
0	2089
3	1167
4	754
1	689

The five cluster we get from the model

```
cluster_0 = df_cp[df_cp["Cluster"] == 0]
cluster_0.head()
```

	Card_Category	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon	Contacts_Count
2	Blue	36	4	1	
3	Blue	34	3	4	
4	Blue	21	5	1	

```
cluster_1 = df_cp[df_cp["Cluster"] == 1]
cluster_1.head()
```

	Card_Category	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon	Contacts_Count
6	Gold	46	6	1	
7	Silver	27	2	2	
16	Blue	36	6	2	
40	Blue	41	2	2	
45	Blue	30	3	2	

```
cluster_2 = df_cp[df_cp["Cluster"] == 2]
cluster_2.head()
```

	Card_Category	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon	Contacts_Count
0	Blue	39	5	1	
9	Blue	36	6	3	
12	Blue	36	3	6	
17	Blue	34	4	4	
19	Blue	37	6	1	

```
cluster_3 = df_cp[df_cp["Cluster"] == 3]
cluster_3.head()
```

	Card_Category	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon	Contacts_Count
8	Blue	36	5	2	
20	Blue	42	5	2	
48	Blue	40	4	3	
53	Blue	36	4	2	
63	Blue	32	2	4	

```
cluster_4 = df_cp[df_cp["Cluster"] == 4]
cluster_4.head()
```

	Card_Category	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon	Contacts_Count
1	Blue	44	6	1	
10	Blue	31	5	3	
11	Blue	54	6	2	
13	Blue	30	5	1	
25	Blue	28	6	1	

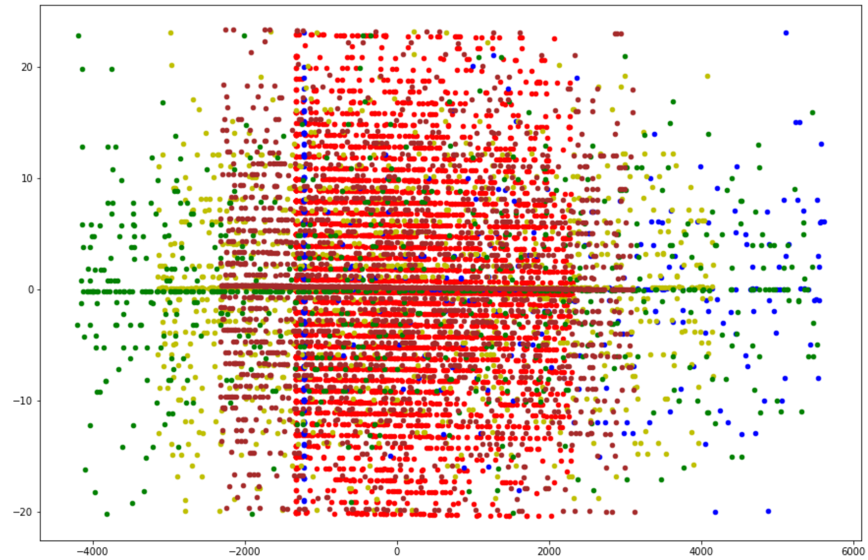
4. Model Insight: K-Means

Use PCA Transform to project customer to 2D plot

```
from sklearn.decomposition import PCA

pca = PCA(n_components=2)

c_0_index = list(cluster_0.index)
c_0 = pca.fit_transform(df_cp_Encoded.loc[c_0_index])

c_1_index = list(cluster_1.index)
c_1 = pca.fit_transform(df_cp_Encoded.loc[c_1_index])

c_2_index = list(cluster_2.index)
```

```
c_2 = pca.fit_transform(df_cp_Encoded.loc[c_2_index])

c_3_index = list(cluster_3.index)
c_3 = pca.fit_transform(df_cp_Encoded.loc[c_3_index])

c_4_index = list(cluster_4.index)
c_4 = pca.fit_transform(df_cp_Encoded.loc[c_4_index])


fig = plt.figure(figsize = [15,10])

plt.scatter(c_0[:,0], c_0[:,1], c = 'r', s = 20, label = 'cluster 0')
plt.scatter(c_1[:,0], c_1[:,1], c = 'b', s = 20, label = 'cluster 1')
plt.scatter(c_2[:,0], c_2[:,1], c = 'y', s = 20, label = 'cluster 2')
plt.scatter(c_3[:,0], c_3[:,1], c = 'g', s = 20, label = 'cluster 3')
plt.scatter(c_4[:,0], c_4[:,1], c = 'brown', s = 20, label = 'cluster 4')
```

```
<matplotlib.collections.PathCollection at 0x7f7894d4b790>
```



Unfortunately, we didn't find clear clusters in the 2D plot. The reason is that K means is good for clustering data that can form a ball in hyper dimension. If we project featuers of our data into 2D plots, we can see that all the data points are nearly uniformly distributed in the 2D plots. They do not have potential to form clusters. That is probably the reason that we cannot form clear clusters from the data set. I also make plots that shows distributions of each pair of features. If you are interested, I put those plots in part 2 of the report.

5. Latent Dirichlet Allocation(LDA) Model

```
from sklearn.decomposition import LatentDirichletAllocation
lda = LatentDirichletAllocation(n_components=5)
lda_output = lda.fit_transform(df_cp_Encoded)

print(lda_output.shape)
```

```
(10127, 5)
```

```
topic_center = lda.components_

print(topic_center.shape)
print(topic_center)
```

```
(5, 6)
[[3.01197831e+02 8.22861075e+02 1.09310814e+02 5.36668001e+01
  5.81933284e+01 1.83393154e+07]
 [3.99281938e+02 2.35207817e+04 2.20439019e+03 1.38230970e+03
  1.72869795e+03 1.74554396e+07]
 [3.43363225e+00 3.24225645e+05 3.47902464e+04 2.09155202e+04
  2.14706550e+04 1.54126596e+07]
 [9.49090759e+02 9.72405815e+03 9.79693481e+02 8.43292047e+02
```

```
 9.55318991e+02 1.80585056e+07]
[1.68995839e+02 5.55465441e+03 5.27359135e+02 5.15211230e+02
 6.53134723e+02 1.81498759e+07]]
```

```
# column names
cluster_number = ["Cluster" + str(i) for i in range(lda.n_components)]

df_document_cluster = pd.DataFrame(np.round(lda_output, 2), columns=cluster_number)

# get dominant topic for each document
cluster = np.argmax(df_document_cluster.values, axis=1)
df_document_cluster['cluster'] = cluster

df_document_cluster.head(15)
```

	Cluster0	Cluster1	Cluster2	Cluster3	Cluster4	cluster
0	0.22	0.21	0.14	0.22	0.22	0
1	0.19	0.19	0.23	0.19	0.19	2
2	0.15	0.16	0.40	0.15	0.15	2
3	0.14	0.15	0.42	0.14	0.14	2
4	0.20	0.20	0.20	0.20	0.20	0
5	0.16	0.17	0.36	0.16	0.16	2
6	0.24	0.22	0.07	0.23	0.24	0
7	0.25	0.22	0.05	0.24	0.24	0
8	0.24	0.22	0.08	0.23	0.23	0
9	0.22	0.21	0.15	0.21	0.21	0
10	0.20	0.20	0.21	0.20	0.20	2
11	0.19	0.19	0.25	0.19	0.19	2
12	0.22	0.21	0.14	0.21	0.22	0
13	0.21	0.21	0.16	0.21	0.21	0
14	0.07	0.09	0.69	0.08	0.08	2

```
cluster_number = [i for i in range(lda.n_components)]
df_document_cluster = pd.DataFrame(np.round(lda_output, 2), columns=cluster_number)
cluster = np.argmax(df_document_cluster.values, axis=1)
df_cp['Cluster'] = cluster
df_cp.head(10)
```

```
<ipython-input-14-63f1fadb831a>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide
  df_cp['Cluster'] = cluster
```

	Card_Category	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mo
0	Blue	39	5	
1	Blue	44	6	
2	Blue	36	4	
3	Blue	34	3	
4	Blue	21	5	
5	Blue	36	3	
6	Gold	46	6	
7	Silver	27	2	
8	Blue	36	5	
9	Blue	36	6	

```
cluster_0 = df_cp[df_cp["Cluster"] == 0]
cluster_1 = df_cp[df_cp["Cluster"] == 1]
cluster_2 = df_cp[df_cp["Cluster"] == 2]
cluster_3 = df_cp[df_cp["Cluster"] == 3]
cluster_4 = df_cp[df_cp["Cluster"] == 4]
```

```
print ("Number of reviews included in each cluster:")
df_cp['Cluster'].value_counts().to_frame()
```

Number of reviews included in each cluster:

	Cluster
2	6560
0	3480
3	84
4	3

We can see that cluster 1 and cluster 2, as well as cluster 3 and cluster 4 are always having similar probabilities. They may be similar clusters. Besides, there is no element in cluster 1. As a result, in this case, 5 may not be a good cluster number. Let's try other cluster numbers.

```
lda2 = LatentDirichletAllocation(n_components=3)
lda2_output = lda2.fit_transform(df_cp_Encoded)

# column names
cluster_number = ["Cluster" + str(i) for i in range(lda2.n_components)]

df_document_cluster = pd.DataFrame(np.round(lda2_output, 2), columns=cluster_number)

# get dominant topic for each document
cluster = np.argmax(df_document_cluster.values, axis=1)
df_document_cluster['cluster'] = cluster

df_document_cluster.head(10)
```

	Cluster0	Cluster1	Cluster2	cluster
0	0.38	0.37	0.25	0
1	0.32	0.32	0.36	2
2	0.22	0.23	0.55	2
3	0.21	0.22	0.57	2
4	0.33	0.34	0.33	1
5	0.24	0.26	0.50	2
6	0.43	0.40	0.16	0
7	0.45	0.41	0.14	0
8	0.42	0.40	0.18	0
9	0.37	0.36	0.27	0

```
cluster_number = [i for i in range(lda2.n_components)]
df_document_cluster = pd.DataFrame(np.round(lda2_output, 2), columns=cluster_number)
cluster = np.argmax(df_document_cluster.values, axis=1)
df_cp['Cluster'] = cluster
df_cp.head(10)
```

```
<ipython-input-36-cd572c7e8d63>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide
  df_cp['Cluster'] = cluster
```

	Card_Category	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mo
0	Blue	39	5	
1	Blue	44	6	
2	Blue	36	4	
3	Blue	34	3	
4	Blue	21	5	
5	Blue	36	3	
6	Gold	46	6	
7	Silver	27	2	
8	Blue	36	5	
9	Blue	36	6	

```
print ("Number of reviews included in each cluster:")
df_cp['Cluster'].value_counts().to_frame()
```

Number of reviews included in each cluster:

	Cluster
2	6552
0	3573
1	2

Similar to the first case where cluster number is 5, there is a cluster that is very small. In this case, we can change cluster number to 2.

```
lda3 = LatentDirichletAllocation(n_components=2)
lda3_output = lda3.fit_transform(df_cp_Encoded)

# column names
cluster_number = ["Cluster" + str(i) for i in range(lda3.n_components)]

df_document_cluster = pd.DataFrame(np.round(lda3_output, 2), columns=cluster_number)

# get dominant topic for each document
cluster = np.argmax(df_document_cluster.values, axis=1)
df_document_cluster['cluster'] = cluster

df_document_cluster.head(10)
```

	Cluster0	Cluster1	cluster
0	0.57	0.43	0
1	0.47	0.53	1
2	0.31	0.69	1
3	0.29	0.71	1
4	0.49	0.51	1
5	0.34	0.66	1
6	0.66	0.34	0
7	0.68	0.32	0
8	0.65	0.35	0
9	0.55	0.45	0

```
df_cp['Cluster'] = cluster
df_cp.head(10)
```

```
<ipython-input-42-e58de5a68fb7>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide
  df_cp['Cluster'] = cluster
```

	Card_Category	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mo
0	Blue	39	5	
1	Blue	44	6	
2	Blue	36	4	
3	Blue	34	3	
4	Blue	21	5	
5	Blue	36	3	
6	Gold	46	6	
7	Silver	27	2	
8	Blue	36	5	
9	Blue	36	6	

```
print ("Number of reviews included in each cluster:")
df_cp['Cluster'].value_counts().to_frame()
```

Number of reviews included in each cluster:

	Cluster
1	6627
0	3500

6. Model Insight: LDA

```
cluster_0 = df_cp[df_cp["Cluster"] == 0]
cluster_1 = df_cp[df_cp["Cluster"] == 1]


from sklearn.decomposition import PCA

pca = PCA(n_components=2)

c_0_index = list(cluster_0.index)
c_0 = pca.fit_transform(df_cp_Encoded.loc[c_0_index])

c_1_index = list(cluster_1.index)
c_1 = pca.fit_transform(df_cp_Encoded.loc[c_1_index])


fig = plt.figure(figsize = [15,10])

plt.scatter(c_0[:,0], c_0[:,1], c = 'r', s = 20, label = 'cluster 0')
plt.scatter(c_1[:,0], c_1[:,1], c = 'g', s = 20, label = 'cluster 1')
```

<matplotlib.collections.PathCollection at 0x7f788e2673d0>