7

April 17, 2023

1 Group Name: Go Bear!

1.1 Name: Xiaoke Song

1.2 Email: xiaokesong57@gmail.com

1.3 Country: born in China, college in the US

1.4 College: UC Berkeley

1.5 Specialization: Data Science

Problem description: Customer Segmation _____ "XYZ bank wants to roll out Christmas offers to their customers. But Bank does not want to roll out same offer to all customers instead they want to roll out personalized offer to particular set of customers. If they manually start understanding the category of customer then this will be not efficient and also they will not be able to uncover the hidden pattern in the data (pattern which group certain kind of customer in one category). Bank approached ABC analytics company to solve their problem. Bank also shared information with ABC analytics that they don't want more than 5 group as this will be inefficient for their campaign."

Business understanding: The business problem is to segment the customers of XYZ bank into distinct groups based on their behavior patterns so that personalized offers can be rolled out for the Christmas campaign. The bank wants to identify no more than five groups of customers. The objective of this project is to come up with an approach to segment the customers and provide recommendations to the bank.

Project lifecycle along with deadline: The project will be completed in four weeks.

Data Understanding: The data contains information about customers of the bank, including demographic information, customer behavior, and product ownership. There are several columns in the dataset, including customer code, age, seniority, activity index, gross income, and various product ownership indicators.

EDA: Exploratory Data Analysis involves examining the data to identify patterns, trends, and relationships that can inform the customer segmentation approach. We can perform descriptive statistics, data visualization, and correlation analysis to better understand the data.

Feature Engineering: Feature engineering involves creating new features from the existing dataset to improve the model's accuracy. We can create new variables such as customer lifetime value, purchase frequency, and average transaction value to better understand customer behavior and segment them accordingly.

Model Building: We can use clustering algorithms such as K-Means, Hierarchical Clustering, or DBSCAN to segment the customers into distinct groups based on their behavior patterns. The number of clusters can be determined using techniques such as the Elbow Method, Silhouette Score, or Gap Statistic.

Model Evaluation: We can evaluate the performance of the clustering algorithm by examining metrics such as Within Cluster Sum of Squares (WCSS), Silhouette Score, and Adjusted Rand Index (ARI).

Presentation: We will present the customer segmentation approach to the bank in a recommendation slide that includes a summary of the approach, the identified customer groups, and recommendations for personalized offers for each group.

Data Intake report:

[1]:
```
import pandas as pd
```

[7]:
```
df = pd.read_csv("cust_seg.csv")
df.head(5)
```

```
/var/folders/_0/nmpfpzw134n12j0c0z6jtrw80000gn/T/ipykernel_25294/1520097819.py:1
: DtypeWarning: Columns (16) have mixed types. Specify dtype option on import or
set low_memory=False.
  df = pd.read_csv("cust_seg.csv")
```

[7]:

	Unnamed: 0	fecha_dato	ncodpers	ind_empleado	pais_residencia	sexo	age	\
0	0	2015-01-28	1375586	N	ES	H	35	
1	1	2015-01-28	1050611	N	ES	V	23	
2	2	2015-01-28	1050612	N	ES	V	23	
3	3	2015-01-28	1050613	N	ES	H	22	
4	4	2015-01-28	1050614	N	ES	V	23	

	fecha_alta	ind_nuevo	antiguedad	…	ind_hip_fin_ult1	ind_plan_fin_ult1	\
0	2015-01-12	0.0	6	…	0	0	
1	2012-08-10	0.0	35	…	0	0	
2	2012-08-10	0.0	35	…	0	0	
3	2012-08-10	0.0	35	…	0	0	
4	2012-08-10	0.0	35	…	0	0	

	ind_pres_fin_ult1	ind_reca_fin_ult1	ind_tjcr_fin_ult1	ind_valo_fin_ult1	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	ind_viv_fin_ult1	ind_nomina_ult1	ind_nom_pens_ult1	ind_recibo_ult1
0	0	0.0	0.0	0
1	0	0.0	0.0	0

```
2                0            0.0             0.0              0
3                0            0.0             0.0              0
4                0            0.0             0.0              0

[5 rows x 48 columns]
```

[8]:
```
df.describe()
```

[8]:
```
          Unnamed: 0      ncodpers      ind_nuevo         indrel  \
count  1000000.000000  1.000000e+06  989218.000000  989218.000000
mean    499999.500000  6.905967e+05       0.000489       1.109074
std     288675.278933  4.044084e+05       0.022114       3.267624
min          0.000000  1.588900e+04       0.000000       1.000000
25%     249999.750000  3.364110e+05       0.000000       1.000000
50%     499999.500000  6.644760e+05       0.000000       1.000000
75%     749999.250000  1.074511e+06       0.000000       1.000000
max     999999.000000  1.379131e+06       1.000000      99.000000

         indrel_1mes   tipodom       cod_prov  ind_actividad_cliente  \
count  989218.000000  989218.0  982266.000000          989218.000000
mean        1.000085       1.0      26.852131               0.564971
std         0.012954       0.0      12.422924               0.495761
min         1.000000       1.0       1.000000               0.000000
25%         1.000000       1.0      18.000000               0.000000
50%         1.000000       1.0      28.000000               1.000000
75%         1.000000       1.0      33.000000               1.000000
max         3.000000       1.0      52.000000               1.000000

              renta  ind_ahor_fin_ult1  …  ind_hip_fin_ult1  \
count  8.248170e+05     1000000.000000  …     1000000.000000
mean   1.396462e+05           0.000177  …           0.009982
std    2.389858e+05           0.013303  …           0.099410
min    1.202730e+03           0.000000  …           0.000000
25%    7.157184e+04           0.000000  …           0.000000
50%    1.066519e+05           0.000000  …           0.000000
75%    1.634325e+05           0.000000  …           0.000000
max    2.889440e+07           1.000000  …           1.000000

       ind_plan_fin_ult1  ind_pres_fin_ult1  ind_reca_fin_ult1  \
count     1000000.000000     1000000.000000     1000000.000000
mean            0.014553           0.004661           0.072581
std             0.119755           0.068112           0.259448
min             0.000000           0.000000           0.000000
25%             0.000000           0.000000           0.000000
50%             0.000000           0.000000           0.000000
75%             0.000000           0.000000           0.000000
max             1.000000           1.000000           1.000000
```

```
       ind_tjcr_fin_ult1  ind_valo_fin_ult1  ind_viv_fin_ult1  \
count     1000000.000000     1000000.000000    1000000.000000
mean            0.066084           0.039378          0.006442
std             0.248429           0.194493          0.080003
min             0.000000           0.000000          0.000000
25%             0.000000           0.000000          0.000000
50%             0.000000           0.000000          0.000000
75%             0.000000           0.000000          0.000000
max             1.000000           1.000000          1.000000

       ind_nomina_ult1  ind_nom_pens_ult1  ind_recibo_ult1
count    994598.000000      994598.000000   1000000.000000
mean          0.071629           0.079543         0.166275
std           0.257873           0.270584         0.372327
min           0.000000           0.000000         0.000000
25%           0.000000           0.000000         0.000000
50%           0.000000           0.000000         0.000000
75%           0.000000           0.000000         0.000000
max           1.000000           1.000000         1.000000

[8 rows x 33 columns]
```

Based on the given df.describe() output, the dataset contains 1,000,000 rows and 33 columns. The column names include Unnamed: 0, ncodpers, ind_nuevo, indrel, indrel_1mes, tipodom, cod_prov, ind_actividad_cliente, renta, and 24 other columns whose names are not provided.

The "count" row shows that some columns have missing data, such as ind_nuevo, cod_prov, and renta. The "mean" row provides the average value for each column, while the "std" row shows the standard deviation of each column.

The minimum and maximum values for each column are also provided in the "min" and "max" rows, respectively. The 25th, 50th (median), and 75th percentiles for each column are shown in the rows labeled "25%", "50%", and "75%", respectively.