Customer Segmentation Project

Week 9

Done By: Shaimaa Al-Khawlani

JULY 18, 2022

1. Problem description

Most banks around the world have variant large customer base with different income levels, ages, characteristics, values and lifestyles.

XYZ bank wants to increase the production and the satisfactions of all customers categories by 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).

2. Data Understanding

The existing data, which was provided by the bank, is the bank's customers data. However, the data contains many columns that will help the analytics team analyze the data and build a customer segmentation approach for the bank.

Since the data does not contain a dependent variable or (Target), We believe that machine learning (clustering) techniques would be appropriate to use for this type of data.

Size: 1000000 records, 48 columns.

- **Columns Description:**

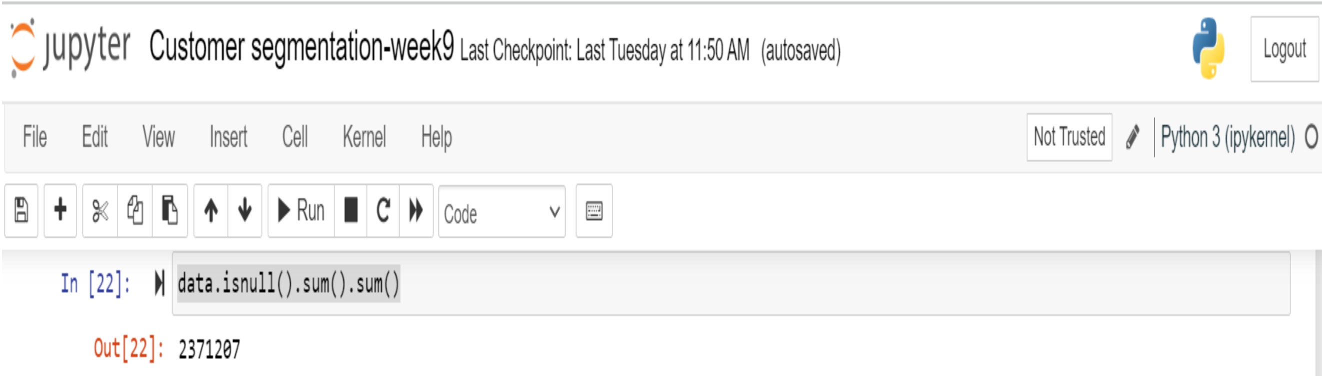
Column Name	Description
fecha_dato	The table is partitioned for this column
ncodpers	Customer code
ind_empleado	Employee index: A active, B ex employed, F filial, N not employee, P pasive
pais_residencia	Customer's Country residence
sexo	Customer's sex
age	Age
fecha_alta	The date in which the customer became as the first holder of a contract in the bank
ind_nuevo	New customer Index. 1 if the customer registered in the last 6 months.
antiguedad	Customer seniority (in months)

indrel	1 (First/Primary), 99 (Primary customer during the month but not at the end of the month)
ult_fec_cli_1t	Last date as primary customer (if he isn't at the end of the month)
indrel_1mes	Customer type at the beginning of the month ,1 (First/Primary customer), 2 (co-owner),P (Potential),3 (former primary), 4(former co-owner)
tiprel_1mes	Customer relation type at the beginning of the month, A (active), I (inactive), P (former customer),R (Potential)
indresi	Residence index (S (Yes) or N (No) if the residence country is the same than the bank country)
indext	Foreigner index (S (Yes) or N (No) if the customer's birth country is different than the bank country)
conyuemp	Spouse index. 1 if the customer is spouse of an employee
canal_entrada	channel used by the customer to join
indfall	Deceased index. N/S
tipodom	Addres type. 1, primary address
cod_prov	Province code (customer's address)
nomprov	Province name
ind_actividad_cliente	Activity index (1, active customer; 0, inactive customer)
renta	Gross income of the household
ind_ahor_fin_ult1	Saving Account
ind_aval_fin_ult1	Guarantees
ind_cco_fin_ult1	Current Accounts
ind_cder_fin_ult1	Derivada Account
ind_cno_fin_ult1	Payroll Account
ind_ctju_fin_ult1	Junior Account
ind_ctma_fin_ult1	Más particular Account
ind_ctop_fin_ult1	particular Account
ind_ctpp_fin_ult1	particular Plus Account
ind_deco_fin_ult1	Short-term deposits
ind_deme_fin_ult1	Medium-term deposits
ind_dela_fin_ult1	Long-term deposits
ind_ecue_fin_ult1	e-account
ind_fond_fin_ult1	Funds
ind_hip_fin_ult1	Mortgage
ind_plan_fin_ult1	Pensions
ind_pres_fin_ult1	Loans
ind_reca_fin_ult1	Taxes
ind_tjcr_fin_ult1	Credit Card

ind_valo_fin_ult1	Securities
ind_viv_fin_ult1	Home Account
ind_nomina_ult1	Payroll
ind_nom_pens_ult1	Pensions
ind_recibo_ult1	Direct Debit

- **Data Cleaning:**
  1. **Handling Missing Values**

From the below print screen we have 2371207 missing values that must be filled or removed.


jupyter
Customer segmentation-week9 Last Checkpoint: Last Tuesday at 11:50 AM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [22]:
data.isnull().sum().sum()
Out[22]:
2371207

jupyter Customer segmentation-week9 Last Checkpoint: Last Tuesday at 11:50 AM (autosaved)

Logout

File Edit View Insert Cell Kernel Help

Not Trusted | Python 3 (ipykernel)

Run Code

In [24]:
```
data.isnull().sum()
```

Out[24]:
```
Unnamed: 0                                                0
data_date                                                 0
customer_code                                             0
employee_index                                        10782
customer_country_residence                            10782
customer_gender                                       10786
age                                                       0
bank_entry_date                                       10782
new_customer_index                                    10782
customer_seniority                                        0
first/primary_customer                                10782
last_date_as_primary_customer                        998899
customer_type_at_the_ beginning_of_the_month          10782
customer_relation_type_at_the_beginning_of_the_ month 10782
residence_index                                       10782
foreign_index                                         10782
spouse_index                                         999822
type_of_channel                                       10861
deceased_index_(N/S)                                  10782
addres_type                                           10782
province_code                                         17734
province_name                                         17734
activity_index                                        10782
gross_income_of_the_ household                       175183
saving_account                                            0
guarantees                                                0
current_account                                           0
derivative_account                                        0
payroll_account                                           0
junior_account                                            0
mas_particular_account                                    0
particular_account                                        0
particular_plus_account                                   0
short_term_deposits                                       0
medium_term_deposits                                      0
long_term_deposits                                        0
e-account                                                 0
funds                                                     0
mortgage                                                  0
pensions                                                  0
loans                                                     0
taxes                                                     0
credit_card                                               0
securities                                                0
home_account                                              0
payroll                                                5402
pensions.1                                             5402
direct_debit                                              0
dtype: int64
```

1 Drop column Unnamed

The missing values contains (NaN and NA) values as the below print screens

yter Customer segmentation-week9 Last Checkpoint: Last Tuesday at 11:50 AM (autosaved)

Logo

Edit View Insert Cell Kernel Help

Not Trusted

Python 3 (ipykernel)

Run Code

[22]:

```
data.isnull().sum().sum()
```

Out[22]: 2371207

[25]:

```
#double check for missin values
pd.set_option('display.max_columns',None)
data
```

Out[25]:

new_customer_index	customer_seniority	first/primary_customer	last_date_as_primary_customer	customer_type_at_the_ beginning_of_the_month	customer_relation_type_at_the_beg
0.0	6	1.0	NaN	1.0	
0.0	35	1.0	NaN	1.0	
0.0	35	1.0	NaN	1.0	
0.0	35	1.0	NaN	1.0	
0.0	35	1.0	NaN	1.0	
...	...	...	...	...	
0.0	22	1.0	NaN	1.0	
0.0	22	1.0	NaN	1.0	
0.0	22	1.0	NaN	1.0	
0.0	22	1.0	NaN	1.0	
0.0	22	1.0	NaN	1.0	

pyter Customer segmentation-week9 Last Checkpoint: Last Tuesday at 11:50 AM (autosaved)

Logout

Edit View Insert Cell Kernel Help

Not Trusted

Python 3 (ipykernel)

Run Code

```
data = data.drop('Unnamed: 0',axis=1)
```

3. Detect other values like "NA"

n [51]:

```
data[data['age'].str.contains('NA')]
```

Out[51]:

	data_date	customer_code	employee_index	customer_country_residence	customer_gender	age	bank_entry_date	new_customer_index	customer_
261	2015-01-28	1050741	NaN	NaN	NaN	NA	NaN	NaN	
1029	2015-01-28	1051017	NaN	NaN	NaN	NA	NaN	NaN	
1063	2015-01-28	1051064	NaN	NaN	NaN	NA	NaN	NaN	
1154	2015-01-28	1051387	NaN	NaN	NaN	NA	NaN	NaN	
1779	2015-01-28	1048660	NaN	NaN	NaN	NA	NaN	NaN	
...	...	...	...	...	...	...	...	...	
998780	2015-02-28	1148591	NaN	NaN	NaN	NA	NaN	NaN	
999111	2015-02-28	1148971	NaN	NaN	NaN	NA	NaN	NaN	
999115	2015-02-28	1148968	NaN	NaN	NaN	NA	NaN	NaN	
999577	2015-02-28	1147591	NaN	NaN	NaN	NA	NaN	NaN	
999724	2015-02-28	1148181	NaN	NaN	NaN	NA	NaN	NaN	

10782 rows × 47 columns

n [55]:

```
df_NA = data[data['customer_seniority'].str.contains('NA')]
df_NA
```

Out[55]:

	data_date	customer_code	employee_index	customer_country_residence	customer_gender	age	bank_entry_date	new_customer_index	customer_
261	2015-01-28	1050741	NaN	NaN	NaN	NA	NaN	NaN	
1029	2015-01-28	1051017	NaN	NaN	NaN	NA	NaN	NaN	
1063	2015-01-28	1051064	NaN	NaN	NaN	NA	NaN	NaN	
1154	2015-01-28	1051387	NaN	NaN	NaN	NA	NaN	NaN	
1779	2015-01-28	1048660	NaN	NaN	NaN	NA	NaN	NaN	
...	...	...	...	...	...	...	...	...	

The above print screen shows that there are 10782 row that are considered as null, so it is recommend to remove them. We retrieved these values based on "NA" Columns of "Age" and "Customer Stationary". There is no need for these row as all of them not indicate and benefit sight so they have been removed.

In [70]:

```
data.isnull().sum()
```

Out[70]:

```
data_date                                                0
customer_code                                            0
employee_index                                           0
customer_country_residence                               0
customer_gender                                          4
age                                                      0
bank_entry_date                                          0
new_customer_index                                       0
customer_seniority                                       0
first/primary_customer                                   0
last_date_as_primary_customer                       988117
customer_type_at_the_ beginning_of_the_month             0
customer_relation_type_at_the_beginning_of_the_ month    0
residence_index                                          0
foreign_index                                            0
spouse_index                                        989040
type_of_channel                                         79
deceased_index_(N/S)                                     0
addres_type                                              0
province_code                                         6952
province_name                                         6952
activity_index                                           0
gross_income_of_the_ household                      164401
saving_account                                           0
guarantees                                               0
current_account                                          0
derivative_account                                       0
payroll_account                                          0
junior_account                                           0
mas_particular_account                                   0
particular_account                                       0
particular_plus_account                                  0
short_term_deposits                                      0
medium_term_deposits                                     0
long_term_deposits                                       0
e-account                                                0
funds                                                    0
mortgage                                                 0
pensions                                                 0
loans                                                    0
taxes                                                    0
credit_card                                              0
securities                                               0
home_account                                             0
payroll                                                100
pensions.1                                             100
direct_debit                                             0
dtype: int64
```

In [71]:

```
data.isnull().sum().sum()
```

Out[71]:

```
2155745
```

After Removing the 10782 rows, the null values decrease significantly from 2371207 to 2155745

Then we replaced the rest of missing values with ZEROS to git ride of all missing values as the below print screen.

```
In [76]:  fill_data.isnull().sum()

Out[76]:  data_date                                              0
          customer_code                                          0
          employee_index                                         0
          customer_country_residence                             0
          customer_gender                                        0
          age                                                    0
          bank_entry_date                                        0
          new_customer_index                                     0
          customer_seniority                                     0
          first/primary_customer                                 0
          last_date_as_primary_customer                          0
          customer_type_at_the_ beginning_of_the_month           0
          customer_relation_type_at_the_beginning_of_the_ month  0
          residence_index                                        0
          foreign_index                                          0
          spouse_index                                           0
          type_of_channel                                        0
          deceased_index_(N/S)                                   0
          addres_type                                            0
          province_code                                          0
          province_name                                          0
          activity_index                                         0
          gross_income_of_the_ household                         0
          saving_account                                         0
          guarantees                                             0
          current_account                                        0
          derivative_account                                     0
          payroll_account                                        0
          junior_account                                         0
          mas_particular_account                                 0
          particular_account                                     0
          particular_plus_account                                0
          short_term_deposits                                    0
          medium_term_deposits                                   0
          long_term_deposits                                     0
          e-account                                              0
          funds                                                  0
          mortgage                                               0
          pensions                                               0
          loans                                                  0
          taxes                                                  0
          credit_card                                            0
          securities                                             0
          home_account                                           0
          payroll                                                0
          pensions.1                                             0
          direct_debit                                           0
          dtype: int64

In [77]:  fill_data.isnull().sum().sum()

Out[77]:  0

In [ ]:
```

Then convert the types of some columns in order to get best insights:

1. Convert Dates columns from Object to Date.

```
In [101]:  for column in ["data_date", "bank_entry_date","last_date_as_primary_customer","last_date_as_primary_customer"] :

                data[column] = pd.to_datetime

            print("\nFeature's datatypes\n\n{}".format(data.dtypes))
```

2. Convert the below columns from Object to integer.

In [95]:
```
#Convert the below columns from Object to integer
for column in ["customer_seniority", "new_customer_index","first/primary_customer","activity_index","province_code"] :

      data[column] = data[column].astype('int64')

print("\nFeature's datatypes\n\n{}".format(data.dtypes))
```

3. Check unique values to convert that needs to convert to category type.

Check Unuique Values to convert that needs to convert to category type

In [88]:
```
#The below table highlights a couple of items that will help determine which values should be categorical.

unique_counts = pd.DataFrame.from_records([(col, data[col].nunique()) for col in data.columns],
                          columns=['Column_Name', 'Num_Unique']).sort_values(by=['Num_Unique'])
unique_counts
```

Out[88]:

Out[88]:

	Column_Name	Num_Unique
18	addres_type	1
0	data_date	2
26	derivative_account	2
27	payroll_account	2
28	junior_account	2
29	mas_particular_account	2
30	particular_account	2
31	particular_plus_account	2
32	short_term_deposits	2
33	medium_term_deposits	2
25	current_account	2
34	long_term_deposits	2
36	funds	2
37	mortgage	2
38	pensions	2
39	loans	2
40	taxes	2
41	credit_card	2
42	securities	2
43	home_account	2
44	payroll	2
35	e-account	2
24	guarantees	2
23	saving_account	2
46	direct_debit	2
21	activity_index	2
17	deceased_index_(N/S)	2
7	new_customer_index	2
14	foreign_index	2
13	residence_index	2
45	pensions.1	2
9	first/primary_customer	2
4	customer_gender	3
15	spouse_index	3
12	customer_relation_type_at_the_beginning_of_the...	3
11	customer_type_at_the_ beginning_of_the_month	3
2	employee_index	5
10	last_date_as_primary_customer	23
19	province_code	53
20	province_name	53
3	customer_country_residence	113

Check unique values of the columns that have object type

In [76]:
```
data['customer_country_residence'].unique()
```

Out[76]:
```
array(['ES', 'CA', 'CH', 'CL', 'IE', 'AT', 'NL', 'FR', 'GB', 'DE', 'DO',
       'BE', 'AR', 'VE', 'US', 'MX', 'BR', 'IT', 'EC', 'PE', 'CO', 'HN',
       'FI', 'SE', 'AL', 'PT', 'MZ', 'CN', 'TW', 'PL', 'IN', 'CR', 'NI',
       'HK', 'AD', 'CZ', 'AE', 'MA', 'GR', 'PR', 'RO', 'IL', 'RU', 'GT',
       'GA', 'NO', 'SN', 'MR', 'UA', 'BG', 'PY', 'EE', 'SV', 'ET', 'CM',
       'SA', 'CI', 'QA', 'LU', 'PA', 'BA', 'BO', 'AU', 'BY', 'KE', 'SG',
       'HR', 'MD', 'SK', 'TR', 'AO', 'CU', 'GQ', 'EG', 'ZA', 'DK', 'UY',
       'GE', 'TH', 'DZ', 'LB', 'JP', 'NG', 'PK', 'TN', 'TG', 'KR', 'GH',
       'RS', 'VN', 'PH', 'KW', 'NZ', 'MM', 'KH', 'GI', 'SL', 'GN', 'GW',
       'OM', 'CG', 'LV', 'LT', 'ML', 'MK', 'HU', 'IS', 'LY', 'CF', 'GM',
       'KZ', 'CD', 'BZ'], dtype=object)
```

In [77]:
```
data['customer_gender'].unique()
```

Out[77]:
```
array(['H', 'V', 0], dtype=object)
```

In [79]:
```
data['customer_type_at_the_ beginning_of_the_month'].unique()
```

Out[79]:
```
array([1, 3, 2], dtype=int64)
```

In [80]:
```
data['customer_relation_type_at_the_beginning_of_the_ month'].unique()
```

Out[80]:
```
array(['A', 'I', 'P'], dtype=object)
```

In [81]:
```
data['residence_index'].unique()
```

Out[81]:
```
array(['S', 'N'], dtype=object)
```

In [82]:
```
data['foreign_index'].unique()
```

Out[82]:
```
array(['N', 'S'], dtype=object)
```

In [83]:
```
data['spouse_index'].unique()
```

Out[83]:
```
array([0, 'N', 'S'], dtype=object)
```

In [84]:
```
data['type_of_channel'].unique()
```

Out[84]:
```
array(['KHL', 'KHE', 'KHD', 'KFA', 'KFC', 'KAT', 'KAZ', 'RED', 'KHC',
       'KHK', 'KGN', 'KHM', 'KHO', 'KDH', 'KEH', 'KAD', 'KBG', 0, 'KGC',
       'KHF', 'KFK', 'KHN', 'KHA', 'KAF', 'KGX', 'KFD', 'KAG', 'KFG',
       'KAB', 'KCC', 'KAE', 'KAH', 'KAR', 'KFJ', 'KFL', 'KAI', 'KFU',
       'KAQ', 'KFS', 'KAA', 'KFP', 'KAJ', 'KFN', 'KGV', 'KGY', 'KFF',
       'KAP', 'KDE', 'KFV', '013', 'K00', 'KAK', 'KCK', 'KCL', 'KAY',
       'KBU', 'KDR', 'KAC', 'KDT', 'KCG', 'KDO', 'KDY', 'KBQ', 'KDA',
       'KBO', 'KCI', 'KEC', 'KBZ', 'KES', 'KDX', 'KAS', '007', 'KEU',
       'KCA', 'KAL', 'KDC', 'KAW', 'KCS', 'KCB', 'KDU', 'KDQ', 'KCN',
       'KCM', '004', 'KCH', 'KCD', 'KCE', 'KEV', 'KBL', 'KEA', 'KBH',
       'KDV', 'KFT', 'KEY', 'KAO', 'KEJ', 'KEO', 'KEI', 'KEW', 'KDZ',
       'KBV', 'KBR', 'KBF', 'KDP', 'KCO', 'KCF', 'KCV', 'KAM', 'KEZ',
       'KBD', 'KAN', 'KBY', 'KCT', 'KDD', 'KBW', 'KCU', 'KBX', 'KDB',
       'KBS', 'KBE', 'KCX', 'KBP', 'KBN', 'KEB', 'KDS', 'KEL', 'KDG',
       'KDF', 'KEF', 'KCP', 'KDM', 'KBB', 'KDW', 'KBJ', 'KFI', 'KBM',
       'KEG', 'KEN', 'KEQ', 'KAV', 'KFH', 'KFM', 'KAU', 'KED', 'KFR',
       'KEK', 'KFB', 'KGW', 'KFE', 'KGU', 'KDI', 'KDN', 'KEE', 'KCR',
       'KCQ', 'KEM', 'KCJ'], dtype=object)
```

In [85]:
```
data['province_name'].unique()
```

Out[85]:
```
array(['MALAGA', 'CIUDAD REAL', 'ZARAGOZA', 'TOLEDO', 'LEON', 'GIPUZKOA',
       'CACERES', 'GIRONA', 'ZAMORA', 'BARCELONA', 'SALAMANCA', 'BURGOS',
       'HUESCA', 'NAVARRA', 'AVILA', 'SEGOVIA', 'LUGO', 'LERIDA',
       'MADRID', 'ALICANTE', 'SORIA', 'SEVILLA', 'CANTABRIA',
       'BALEARS, ILLES', 'VALLADOLID', 'PONTEVEDRA', 'VALENCIA', 'TERUEL',
       'CORUÑA, A', 'OURENSE', 'JAEN', 'CUENCA', 'BIZKAIA', 'CASTELLON',
       'RIOJA, LA', 'ALBACETE', 'BADAJOZ', 'MURCIA', 'CADIZ', 'ALMERIA',
       'GUADALAJARA', 'PALENCIA', 'PALMAS, LAS', 'CORDOBA', 'HUELVA',
       'GRANADA', 'ASTURIAS', 'SANTA CRUZ DE TENERIFE', 'MELILLA',
       'TARRAGONA', 'ALAVA', 0, 'CEUTA'], dtype=object)
```

4. Convert to Category type

```
In [94]:  for column in ["province_name", "type_of_channel" , "spouse_index" , "deceased_index_(N/S)",
                      "foreign_index","residence_index",
                      "customer_relation_type_at_the_beginning_of_the_ month",
                      "customer_type_at_the_ beginning_of_the_month", "customer_gender", "customer_country_residence"] :

              data[column] = data[column].astype('category')
          print("\nFeature's datatypes\n\n{}".format(data.dtypes))
```

Print Types after converting process

```
Feature's datatypes

data_date                                               datetime64[ns]
customer_code                                                    int64
employee_index                                                  object
customer_country_residence                                    category
customer_gender                                               category
age                                                             object
bank_entry_date                                         datetime64[ns]
new_customer_index                                               int64
customer_seniority                                               int64
first/primary_customer                                           int64
last_date_as_primary_customer                           datetime64[ns]
customer_type_at_the_ beginning_of_the_month                  category
customer_relation_type_at_the_beginning_of_the_ month         category
residence_index                                               category
foreign_index                                                 category
spouse_index                                                  category
type_of_channel                                               category
deceased_index_(N/S)                                          category
addres_type                                                    float64
province_code                                                  float64
province_name                                                 category
activity_index                                                 float64
gross_income_of_the_ household                                 float64
saving_account                                                   int64
guarantees                                                       int64
current_account                                                  int64
derivative_account                                               int64
payroll_account                                                  int64
junior_account                                                   int64
mas_particular_account                                           int64
particular_account                                               int64
particular_plus_account                                          int64
short_term_deposits                                              int64
medium_term_deposits                                             int64
long_term_deposits                                               int64
e-account                                                        int64
funds                                                            int64
mortgage                                                         int64
pensions                                                         int64
loans                                                            int64
taxes                                                            int64
credit_card                                                      int64
securities                                                       int64
home_account                                                     int64
payroll                                                        float64
pensions.1                                                     float64
direct_debit                                                     int64
dtype: object
```

5. Drop column “addres_type” since it has only one value and it will not affect data insights.

```
In [98]: #drop Undeeded Columns
         data=data.drop("addres_type",axis=1)
```

Finally, after the above process they have been saved to new excel file.

```
In [102]: data.to_csv('custSeg_cleaned.csv')
```