Customer Segmentation Project

Week 9

AUGUST 23, 2022

Contents

1. Group Information

Group Name**:** M.A.S

Specialization: Data Science

Submitted to: Data Glacier canvas platform

Internship Batch: LISUM10: 30

Group Members	**Three members**		
Name	**Email**	**Country**	**Collage/Company**
Moath Mohammed Bin Musallam	moathmusallam@gmail.com	Saudi Arabia	University of East Anglia
Andrew Kojo Mensah-Onumah	kojoakmo@gmail.com	Ghana	Data Glacier
Shaimaa Saleh Obad Al-khawlani	s.khawlany@gmail.com	Yemen	Data Glacier

2. 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).

3. 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

4. What type of data you have got for analysis?

The dataset provided was CSV format. The dataset contains 1000000 rows and 48 columns. The datasets mostly contain numerical and categorical data types.

Since the data has no target value, the unsupervised learning (clustering) is the best algorithm to use for this kind of data.

Fewer categorical columns have higher cardinality, i.e, they have more than 10 categories. Most of the categorical columns are binary. Among the numerical features, only the `renta` variable is continuous. The rest are integers. It is important to note that some of the binary categorical columns are of float data type.

Below, we have attached snapshots of the datasets and its data types.

```
custSeg_ds.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000000 entries, 0 to 999999
Data columns (total 47 columns):
 #   Column                                                  Non-Null Count    Dtype
---  ------                                                  --------------    -----
 0   data_date                                               1000000 non-null  object
 1   customer_code                                           1000000 non-null  int64
 2   employee_index                                          989218 non-null   object
 3   customer_country_residence                              989218 non-null   object
 4   customer_gender                                         989214 non-null   object
 5   age                                                     1000000 non-null  object
 6   bank_entry_date                                         989218 non-null   object
 7   new_customer_index                                      989218 non-null   float64
 8   customer_seniority                                      1000000 non-null  object
 9   first/primary_customer                                  989218 non-null   float64
 10  last_date_as_primary_customer                           1101 non-null     object
 11  customer_type_at_the_ beginning_of_the_month            989218 non-null   float64
 12  customer_relation_type_at_the_beginning_of_the_ month   989218 non-null   object
 13  residence_index                                         989218 non-null   object
 14  foreign_index                                           989218 non-null   object
 15  spouse_index                                            178 non-null      object
 16  type_of_channel                                         989139 non-null   object
 17  deceased_index_(N/S)                                    989218 non-null   object
 18  addres_type                                             989218 non-null   float64
 19  province_code                                           982266 non-null   float64
 20  province_name                                           982266 non-null   object
 21  activity_index                                          989218 non-null   float64
 22  gross_income_of_the_ household                          824817 non-null   float64
 23  saving_account                                          1000000 non-null  int64
 24  guarantees                                              1000000 non-null  int64
 25  current_account                                         1000000 non-null  int64
 26  derivative_account                                      1000000 non-null  int64
 27  payroll_account                                         1000000 non-null  int64
 28  junior_account                                          1000000 non-null  int64
 29  mas_particular_account                                  1000000 non-null  int64
 30  particular_account                                      1000000 non-null  int64
 31  particular_plus_account                                 1000000 non-null  int64
 32  short_term_deposits                                     1000000 non-null  int64
 33  medium_term_deposits                                    1000000 non-null  int64
 34  long_term_deposits                                      1000000 non-null  int64
 35  e-account                                               1000000 non-null  int64
 36  funds                                                   1000000 non-null  int64
 37  mortgage                                                1000000 non-null  int64
 38  pensions                                                1000000 non-null  int64
 39  loans                                                   1000000 non-null  int64
 40  taxes                                                   1000000 non-null  int64
 41  credit_card                                             1000000 non-null  int64
 42  securities                                              1000000 non-null  int64
 43  home_account                                            1000000 non-null  int64
 44  payroll                                                 994598 non-null   float64
 45  pensions                                                994598 non-null   float64
 46  direct_debit                                            1000000 non-null  int64
dtypes: float64(9), int64(23), object(15)
memory usage: 358.6+ MB
```

5. What are the problems in the data (number of NA values, outliers, skewed, etc.)?

We started initial analysis and we can say that the dataset has some following problems:

1. Missing/null data:

```
draw_missing_data_table(custSeg_ds)
```

56]:

	Total	Percent
spouse_index	999822	0.999822
last_date_as_primary_customer	998899	0.998899
gross_income_of_the_ household	175183	0.175183
province_name	17734	0.017734
province_code	17734	0.017734
type_of_channel	10861	0.010861
customer_gender	10786	0.010786
customer_relation_type_at_the_beginning_of_the_ month	10782	0.010782
activity_index	10782	0.010782
addres_type	10782	0.010782
deceased_index_(N/S)	10782	0.010782
foreign_index	10782	0.010782
residence_index	10782	0.010782
customer_type_at_the_ beginning_of_the_month	10782	0.010782
first/primary_customer	10782	0.010782
new_customer_index	10782	0.010782
bank_entry_date	10782	0.010782
customer_country_residence	10782	0.010782
employee_index	10782	0.010782
payroll	5402	0.005402
pensions	5402	0.005402
loans	0	0.000000
e-account	0	0.000000
funds	0	0.000000
mortgage	0	0.000000
pensions	0	0.000000
data_date	0	0.000000
taxes	0	0.000000
credit_card	0	0.000000
securities	0	0.000000
home_account	0	0.000000
medium_term_deposits	0	0.000000
long_term_deposits	0	0.000000
saving_account	0	0.000000
short_term_deposits	0	0.000000
particular_plus_account	0	0.000000
particular_account	0	0.000000
mas_particular_account	0	0.000000
junior_account	0	0.000000
payroll_account	0	0.000000
derivative_account	0	0.000000
current_account	0	0.000000
guarantees	0	0.000000
customer_code	0	0.000000
customer_seniority	0	0.000000
age	0	0.000000
direct_debit	0	0.000000

```
custSeg_ds.isnull().sum().sum()
```

90]:

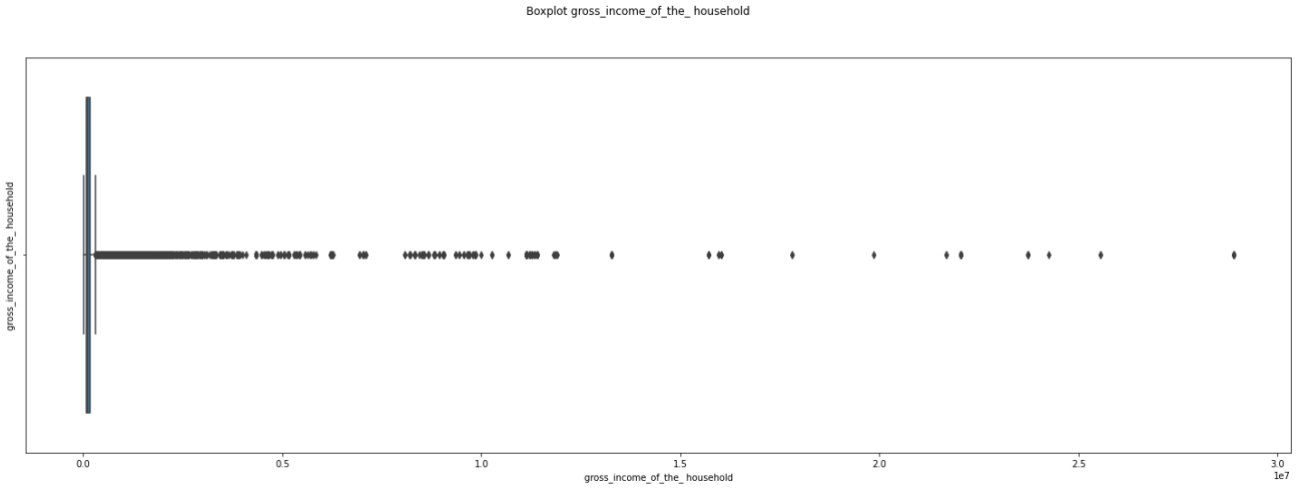
```
2371207
```

From the above analysis we found that the columns have 2371207 missing data, and the columns are listed above.

2. Outliers

```
fig,axes = plt.subplots(figsize=(25, 8), sharey=True)
fig.suptitle('Boxplot gross_income_of_the_ household')
sns.boxplot( x='gross_income_of_the_ household' , data=custSeg_ds ).set_ylabel("gross_income_of_the_ household")
```

```
Text(0, 0.5, 'gross_income_of_the_ household')
```



As per the above there are significant outliers in “gross_income_of_the_ household” , Upper and Lower Limits of gross_income_of_the_ household is:
[-66219.10500000001, 301223.41500000004]

6. Approaches to apply to deal with identified problems

- Columns “spouse index” and “last_date_as_primary_customer” will be dropped because 99.8 percent of their values are missing.
- Imputing missing date values by either using interpolation, backfill or forward fill methods.
- Replacing some missing numerical values by mean, mode or median methods. Also, using KNN imputation technique and other approaches for numerical variables.
- Replacing missing categorical variables with mode and/or by using unsupervised techniques.

7. Data Cleansing and Transformation

Different techniques were used to deal with missing values and outliers in the dataset. Also, several approaches were used in feature transformation. Below I mention the steps used:

- Rename columns with English alternatives.

- Change data types of columns.

- Inspect missing values.

- Drop columns with high number of missing values and default naming column.

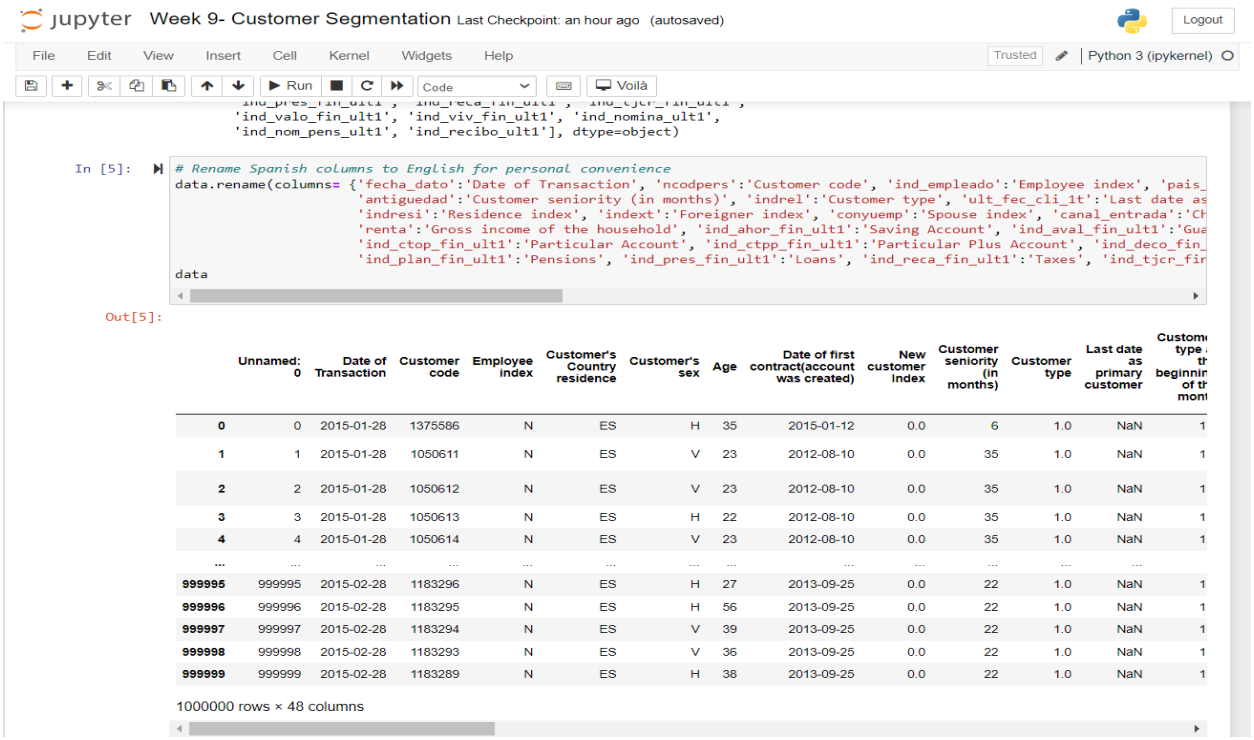
- Inspect skewness and distribution of columns.

- View missing values in dataframe.

- Drop rows with missing values.

- Impute special missing data with column mode.

- Re-inspect dataframe to confirm there are no missing values.



jupyter Week 9- Customer Segmentation Last Checkpoint: an hour ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code Voilà

In [10]:
```
# Set data types of several columns

data['Age'] = pd.to_numeric(data['Age'], errors = 'coerce')
data['Customer seniority (in months)'] = pd.to_numeric(data['Customer seniority (in months)'], errors = 'coerce')
data['Date of Transaction'] = pd.to_datetime(data['Date of Transaction'])
data['Date of first contract(account was created)'] = pd.to_datetime(data['Date of first contract(account was created)'])
data['Customer code'] = data['Customer code'].astype(str)
data['Province code (customer\'s address)'] = data['Province code (customer\'s address)'].astype(str)
data['Customer type'] = data['Customer type'].astype(str)
data['Customer type at the beginning of the month'] = data['Customer type at the beginning of the month'].astype(str)
print(data.dtypes)
```

```
Date of Transaction                                    datetime64[ns]
Customer code                                                  object
Employee index                                                 object
Customer's Country residence                                   object
Customer's sex                                                 object
Age                                                           float64
Date of first contract(account was created)            datetime64[ns]
New customer Index                                            float64
Customer seniority (in months)                                float64
Customer type                                                  object
Customer type at the beginning of the month                    object
Customer relation type at the beginning of the month           object
Residence index                                                object
Foreigner index                                                object
Channel used by the customer to join                           object
Deceased index                                                 object
Addres type                                                   float64
Province code (customer's address)                             object
Province name                                                  object
Activity index                                                float64
Gross income of the household                                 float64
Saving Account                                                  int64
Guarantees                                                      int64
Current Account                                                 int64
Derivada Account                                                int64
Payroll Account                                                 int64
Junior Account                                                  int64
Más particular Account                                          int64
Particular Account                                              int64
Particular Plus Account                                         int64
```

jupyter Week 9- Customer Segmentation Last Checkpoint: an hour ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code Voilà

In [8]:
```
# Check for missing values
data.isnull().sum()
```

Out[8]:
```
Unnamed: 0                                                  0
Date of Transaction                                         0
Customer code                                               0
Employee index                                          10782
Customer's Country residence                            10782
Customer's sex                                          10786
Age                                                         0
Date of first contract(account was created)             10782
New customer Index                                      10782
Customer seniority (in months)                              0
Customer type                                           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
Foreigner index                                         10782
Spouse index                                           999822
Channel used by the customer to join                    10861
Deceased index                                          10782
Addres type                                             10782
Province code (customer's address)                      17734
Province name                                           17734
Activity index                                          10782
Gross income of the household                          175183
Saving Account                                              0
Guarantees                                                  0
Current Account                                             0
Derivada Account                                            0
Payroll Account                                             0
Junior Account                                              0
Más 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
```

jupyter Week 9- Customer Segmentation Last Checkpoint: an hour ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code Voilà

```
Customer type at the beginning of the month              10782
Customer relation type at the beginning of the month     10782
Residence index                                          10782
Foreigner index                                          10782
Spouse index                                            999822
Channel used by the customer to join                     10861
Deceased index                                           10782
Addres type                                              10782
Province code (customer's address)                       17734
Province name                                            17734
Activity index                                           10782
Gross income of the household                           175183
Saving Account                                               0
Guarantees                                                   0
Current Account                                              0
Derivada Account                                             0
Payroll Account                                              0
Junior Account                                               0
Más 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                                                  5402
Direct Debit                                                 0
dtype: int64
```

In [9]:

```
# Drop default column and columns with high number of missing values
data.drop(columns = ['Unnamed: 0', 'Last date as primary customer', 'Spouse index'], inplace = True)
```

jupyter Week 9- Customer Segmentation Last Checkpoint: an hour ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code Voilà

Visualisation for detection (outliers, etc)

In [18]:

```
skewvalue= data.drop_duplicates().skew(axis = 0, skipna = True)
print('Skew value: ', skewvalue)
```

```
Skew value:  Customer code                                    0.032278
Age                                                           0.585771
New customer Index                                           45.175718
Customer seniority (in months)                             -496.496738
Customer type                                                      NaN
Customer type at the beginning of the month                        NaN
Addres type                                                   0.000000
Province code (customer's address)                                 NaN
Activity index                                               -0.262105
Gross income of the household                                52.234130
Saving Account                                               75.144758
Guarantees                                                  160.119026
Current Account                                              -1.152401
Derivada Account                                             41.098090
Payroll Account                                               2.571915
Junior Account                                                8.391620
Más particular Account                                        9.903618
Particular Account                                            1.405709
Particular Plus Account                                       3.309287
Short-term deposits                                          21.456830
Medium-term deposits                                         17.733145
Long-term deposits                                            3.467512
e-account                                                     2.555226
Funds                                                         5.815247
Mortgage                                                      9.858534
Pensions                                                      8.107362
Loans                                                        14.544812
Taxes                                                         3.294845
Credit Card                                                   3.493287
Securities                                                    4.736660
Home Account                                                 12.338486
Payroll                                                       3.322353
Pensions                                                      3.107782
Direct Debit                                                 1.792646
```

jupyter Week 9- Customer Segmentation Last Checkpoint: an hour ago (autosaved)

Logout

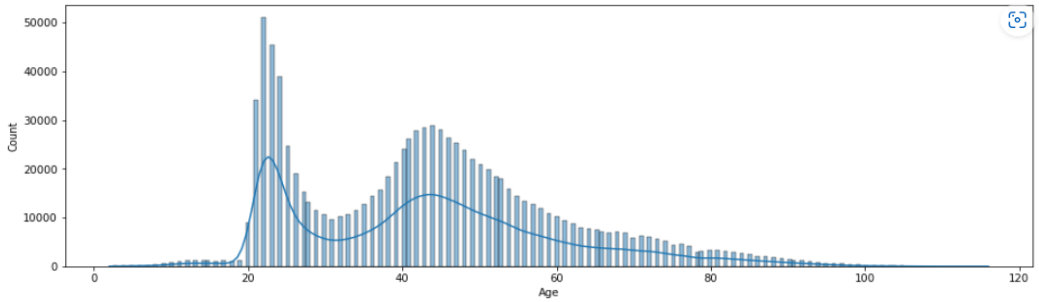
File Edit View Insert Cell Kernel Widgets Help

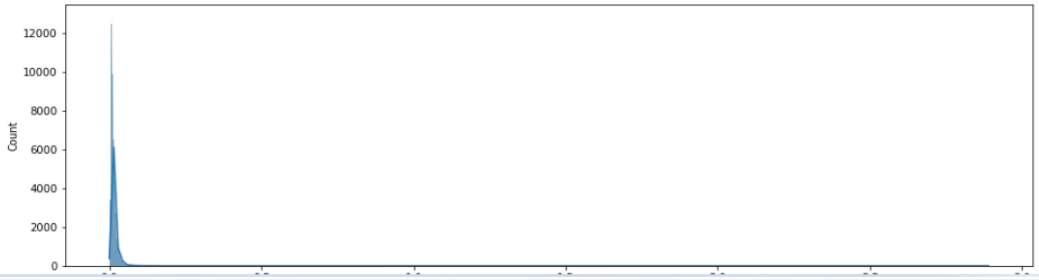
Trusted | Python 3 (ipykernel)

Run Code Voilà

In [19]:
```
# create subplots
fig, ax = plt.subplots(3,1, figsize= (16,16))
sns.histplot(x = data.Age, kde = True, ax= ax[0])
sns.histplot(x = data['Gross income of the household'], kde = True, ax= ax[1])
sns.histplot(x = data['Guarantees'], kde = True, ax= ax[2])
```

Out[19]: <AxesSubplot:xlabel='Guarantees', ylabel='Count'>





jupyter Week 9- Customer Segmentation Last Checkpoint: an hour ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code Voilà

Attempt at dealing with missing values

In [24]:
```
data.describe()
```

Out[24]:

	Age	New customer Index	Customer seniority (in months)	Addres type	Activity index	Gross income of the household	Saving Account	Guarantees	Current Account	Deriv Acc
count	989218.000000	989218.000000	989218.000000	989218.0	989218.000000	8.248170e+05	1000000.000000	1000000.000000	1000000.000000	1000000.00
mean	43.269624	0.000489	93.093975	1.0	0.564971	1.396462e+05	0.000177	0.000039	0.749626	0.00
std	17.158355	0.022114	2012.137351	0.0	0.495761	2.389858e+05	0.013303	0.006245	0.433229	0.02
min	2.000000	0.000000	-999999.000000	1.0	0.000000	1.202730e+03	0.000000	0.000000	0.000000	0.00
25%	27.000000	0.000000	33.000000	1.0	0.000000	7.157184e+04	0.000000	0.000000	0.000000	0.00
50%	43.000000	0.000000	97.000000	1.0	1.000000	1.066519e+05	0.000000	0.000000	1.000000	0.00
75%	53.000000	0.000000	157.000000	1.0	1.000000	1.634325e+05	0.000000	0.000000	1.000000	0.00
max	116.000000	1.000000	246.000000	1.0	1.000000	2.889440e+07	1.000000	1.000000	1.000000	1.00

In [25]:
```
#data[data['Age']==116]
data[data['Customer seniority (in months)']==-999999.0] #this value represents missing data
```

Out[25]:

	Date of Transaction	Customer code	Employee index	Customer's Country residence	Customer's sex	Age	Date of first contract(account was created)	New customer Index	Customer seniority (in months)	Customer type	Customer type at the beginning of the month	Customer relation type at the beginning of the month	Residen ind
452687	2015-01-28	138388	N	ES	V	51.0	1999-07-16	0.0	-999999.0	1.0	1.0	A	
461981	2015-01-28	162278	N	ES	V	66.0	2005-06-08	0.0	-999999.0	1.0	1.0	A	
788786	2015-02-28	162278	N	ES	V	66.0	2005-06-08	0.0	-999999.0	1.0	1.0	A	
800687	2015-02-28	138388	N	ES	V	51.0	1999-07-16	0.0	-999999.0	1.0	1.0	A	

jupyter Week 9- Customer Segmentation Last Checkpoint: an hour ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code Voilà

In [26]:
```
# Replace column with value '999999' with mode
data['Customer seniority (in months)'].mode()
data.replace([-999999.0, 21], inplace = True)
```

In [27]:
```
# Verify that replace function worked
data[data['Customer seniority (in months)']==-999999.0]
```

Out[27]:

Date of Transaction	Customer code	Employee index	Customer's Country residence	Customer's sex	Age	Date of first contract(account was created)	New customer Index	Customer seniority (in months)	Customer type	Customer type at the beginning of the month	Customer relation type at the beginning of the month	Residence index	Fo

In [28]:
```
data.describe()
```

Out[28]:

	Age	New customer Index	Customer seniority (in months)	Addres type	Activity index	Gross income of the household	Saving Account	Guarantees	Current Account	Deriv Acco
count	988453.000000	989218.000000	970553.000000	989218.0	989218.000000	8.248170e+05	1000000.000000	1000000.000000	1000000.000000	1000000.000
mean	43.526559	0.000489	98.535484	1.0	0.564971	1.396462e+05	0.000177	0.000039	0.749626	0.000
std	17.003273	0.022114	65.722077	0.0	0.495761	2.389858e+05	0.013303	0.006245	0.433229	0.024
min	2.000000	0.000000	0.000000	1.0	0.000000	1.202730e+03	0.000000	0.000000	0.000000	0.000
25%	28.000000	0.000000	34.000000	1.0	0.000000	7.157184e+04	0.000000	0.000000	0.000000	0.000
50%	43.000000	0.000000	100.000000	1.0	1.000000	1.066519e+05	0.000000	0.000000	1.000000	0.000
75%	53.000000	0.000000	158.000000	1.0	1.000000	1.634325e+05	0.000000	0.000000	1.000000	0.000
max	116.000000	1.000000	246.000000	1.0	1.000000	2.889440e+07	1.000000	1.000000	1.000000	1.000

In [29]:
```
# Remove any row that has a column with missing value(s)
drop_frame = data.dropna()
drop_frame
```

jupyter Week 9- Customer Segmentation Last Checkpoint: an hour ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code Voilà

In [29]:
```
# Remove any row that has a column with missing value(s)
drop_frame = data.dropna()
drop_frame
```

Out[29]:

	Date of Transaction	Customer code	Employee index	Customer's Country residence	Customer's sex	Age	Date of first contract(account was created)	New customer Index	Customer seniority (in months)	Customer type	Customer type at the beginning of the month	Customer relation type at the beginning of the month	Residen ind
0	2015-01-28	1375586	N	ES	H	35.0	2015-01-12	0.0	6.0	1.0	1.0	A	
1	2015-01-28	1050611	N	ES	V	23.0	2012-08-10	0.0	35.0	1.0	1.0	I	
2	2015-01-28	1050612	N	ES	V	23.0	2012-08-10	0.0	35.0	1.0	1.0	I	
3	2015-01-28	1050613	N	ES	H	22.0	2012-08-10	0.0	35.0	1.0	1.0	I	
5	2015-01-28	1050615	N	ES	H	23.0	2012-08-10	0.0	35.0	1.0	1.0	I	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
999995	2015-02-28	1183296	N	ES	H	27.0	2013-09-25	0.0	22.0	1.0	1.0	A	
999996	2015-02-28	1183295	N	ES	H	56.0	2013-09-25	0.0	22.0	1.0	1.0	A	
999997	2015-02-28	1183294	N	ES	V	39.0	2013-09-25	0.0	22.0	1.0	1.0	A	
999998	2015-02-28	1183293	N	ES	V	36.0	2013-09-25	0.0	22.0	1.0	1.0	A	
999999	2015-02-28	1183289	N	ES	H	38.0	2013-09-25	0.0	22.0	1.0	1.0	A	

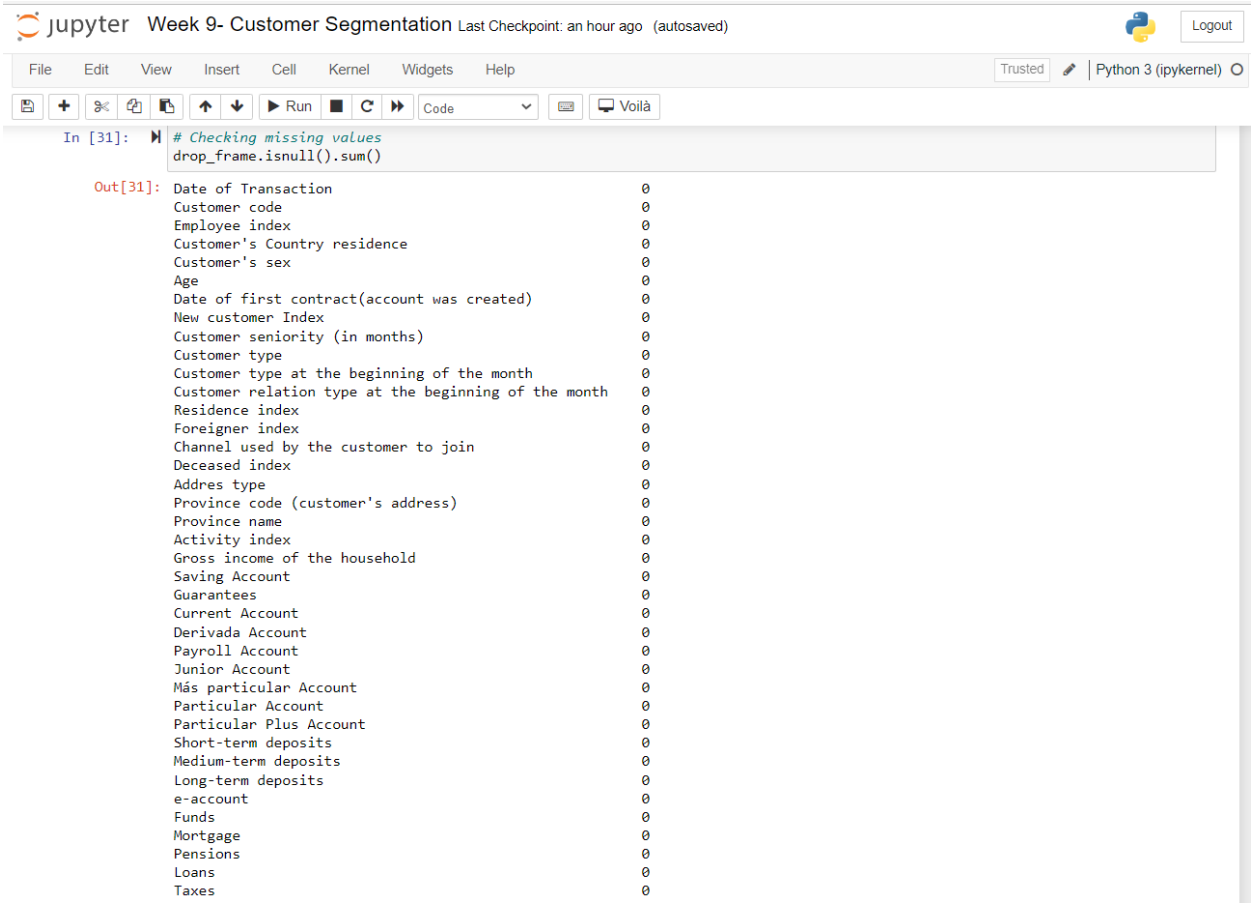
808866 rows × 45 columns

In [30]:
```
# Number of rows dropped
1000000-824680
```

Out[30]: 175320

In [31]:
```
# Checking missing values
drop_frame.isnull().sum()
```

Out[31]:
```
Date of Transaction                          0
```


jupyter Week 9- Customer Segmentation Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Voilà
In [31]:
# Checking missing values
drop_frame.isnull().sum()
Out[31]:
Date of Transaction 0
Customer code 0
Employee index 0
Customer's Country residence 0
Customer's sex 0
Age 0
Date of first contract(account was created) 0
New customer Index 0
Customer seniority (in months) 0
Customer type 0
Customer type at the beginning of the month 0
Customer relation type at the beginning of the month 0
Residence index 0
Foreigner index 0
Channel used by the customer to join 0
Deceased index 0
Addres type 0
Province code (customer's address) 0
Province name 0
Activity index 0
Gross income of the household 0
Saving Account 0
Guarantees 0
Current Account 0
Derivada Account 0
Payroll Account 0
Junior Account 0
Más 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