Customer Segmentation Project
Week 8

Name: Customer Segmentation Project
Report date: 02-Sep-2021
Internship Batch: LISUM02
Specialization: Data Science
Group Name: Data Explorers
GitHub link: https://github.com/joeanton719/Customer-Segmentation-Project

Team member's details:

- **Joseph Antony**
    - Email: joeanton719@gmail.com
    - Country: Turkey
    - Company: UrbanStat
- **Melisa Gozet**
    - Email: mgozet@gmail.com
    - Country: Turkey
    - College: Ankara University Artificial Intelligence Technology PhD student
- **Dilem Unal**
    - Email: diilemunal@gmail.com
    - Country: Turkey
    - College/Company: Istanbul Aydın University Software Engineer Student
- **Aynur Cemre Aka**
    - Email: cemreaka@gmail.com
    - Country: Turkey
    - College: Yaşar University Software Engineering Student

Problem Description

Bank XYZ wants to offer Christmas offers to its customers. However, the bank does not want to offer the same offer to all its customers. Instead, they want to deploy the personalized offer to a particular group of customers. It will not be efficient to manually start understanding the category of the customer and they'll not be able to uncover the hidden pattern in the data. ABC analytics assigned this task to their analytics team and instructed their team to come up with the approach and feature which groups similar behavior customers in one category and others in different categories. There shouldn't be more than 5 groups as this will be inefficient.

Data Understanding

The dataset provided contains a list of the Bank's customers from 1995 to 2015. Each observation is supposed to represent the different attributes belonging to a unique customer. The attributes are related to the customer activities with the bank account and other personal information such as the customer's gender, joining date, where the customer is active, the customer's residence, the bank products utilized by the customer, etc.

Column Name (Spanish)	Column Name (English)	Description
fecha_dato	data_date	The table is partitioned for this column
ncodpers	customer_code	Customer code
ind_empleado	employee_index	Employee index: A active, B ex employed, F filial, N not employee, P pasive
pais_residencia	customer_country_residence	Customer's Country residence
sexo	customer_gender	Customer's sex
age	age	Age
fecha_alta	customer_date_of_entry_into_the_bank	The date in which the customer became as the first holder of a contract in the bank
ind_nuevo	new_customer_index	New customer Index. 1 if the customer registered in the last 6 months.
antiguedad	customer_seniority	Customer seniority (in months)
indrel	first/primary_customer	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	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	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	Customer relation type at the beginning of the month, A (active), I (inactive), P (former customer),R (Potential)
indresi	residence_index	Residence index (S (Yes) or N (No) if the residence country is the same than the bank country)
indext	foreign_index	Foreigner index (S (Yes) or N (No) if the customer's birth country is different than the bank country)
conyuemp	spouse_index	Spouse index. 1 if the customer is spouse of an employee
canal_entrada	type_of_channel	channel used by the customer to join
indfall	deceased_index_(N/S)	Deceased index. N/S
tipodom	addres_type	Addres type. 1, primary address
cod_prov	province_code	Province code (customer's address)
nomprov	province_name	Province name
ind_actividad_cliente	activity_index	Activity index (1, active customer; 0, inactive customer)
renta	gross_income_of_the_household	Gross income of the household
ind_ahor_fin_ult1	saving_account	Saving Account
ind_aval_fin_ult1	guarantees	Guarantees
ind_cco_fin_ult1	current_account	Current Accounts
ind_cder_fin_ult1	derivative_account	Derivada Account
ind_cno_fin_ult1	payroll_account	Payroll Account
ind_ctju_fin_ult1	junior_account	Junior Account
ind_ctma_fin_ult1	más_particular_account	Más particular Account
ind_ctop_fin_ult1	particular_account	particular Account
ind_ctpp_fin_ult1	particular_plus_account	particular Plus Account
ind_deco_fin_ult1	short_term_deposits	Short-term deposits

ind_deme_fin_ult1	medium_term_deposits	Medium-term deposits
ind_dela_fin_ult1	long_term_deposits	Long-term deposits
ind_ecue_fin_ult1	e-account	e-account
ind_fond_fin_ult1	funds	Funds
ind_hip_fin_ult1	mortgage	Mortgage
ind_plan_fin_ult1	pensions	Pensions
ind_pres_fin_ult1	loans	Loans
ind_reca_fin_ult1	taxes	Taxes
ind_tjcr_fin_ult1	credit_card	Credit Card
ind_valo_fin_ult1	securities	Securities
ind_viv_fin_ult1	home_account	Home Account
ind_nomina_ult1	payroll	Payroll
ind_nom_pens_ult1	pensions	Pensions
ind_recibo_ult1	direct_debit	Direct Debit

What type of data have you got for analysis?

The dataset provided to us is of CSV format. It is quite large in size and dimension. The dataset contains 1 million rows and 48 columns, having a size of around approximately 366 MB. The datasets mostly contain numerical and categorical data types. A couple of columns are of date-time format.

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.

```
cust_seg_df.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   customer_date_of_entry_into_the_bank                  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
```

Fig 1: Datasets Features and its Data Types (Before Dedup)

```
cust_seg_df = cust_seg_df.drop("Unnamed: 0", axis=1)
cust_seg_df.head()
```

	data_date	customer_code	employee_index	customer_country_residence	customer_gender	age	customer_date_of_entry_into_the_bank	new_customer_index
0	2015-01-28	1375586	N	ES	H	35	2015-01-12	0.0
1	2015-01-28	1050611	N	ES	V	23	2012-08-10	0.0
2	2015-01-28	1050612	N	ES	V	23	2012-08-10	0.0
3	2015-01-28	1050613	N	ES	H	22	2012-08-10	0.0
4	2015-01-28	1050614	N	ES	V	23	2012-08-10	0.0

5 rows × 47 columns

Fig 2: First five observations and 7 columns of the dataset

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

The dataset is far from perfect. During initial dedup analysis, a significant number of missing values and duplicated observation has been observed. Among the categorical variables, there is higher cardinality for some of the categories, with some categories having only negligible counts compared to other categories. This can potentially make our model more biased towards those categories with higher value counts.

Below, snapshots of the data illustrating the problem are attached.

1. Missing Values

```
#checking percentage of missing values for every variable.
cust_seg_df.isna().sum()/len(cust_seg_df)*100
```

```
data_date                                                 0.000000
employee_index                                            1.115531
customer_country_residence                                1.115531
customer_gender                                           1.115851
age                                                       0.000000
customer_date_of_entry_into_the_bank                      1.115531
new_customer_index                                        1.115531
customer_seniority                                        0.000000
first/primary_customer                                    1.115531
last_date_as_primary_customer                            99.876549
customer_type_at_the_beginning_of_the_month               1.115531
customer_relation_type_at_the_beginning_of_the_month      1.115531
residence_index                                           1.115531
foreign_index                                             1.115531
spouse_index                                             99.985786
type_of_channel                                           1.123676
deceased_index_(N/S)                                      1.115531
addres_type                                               1.115531
province_code                                             1.703561
province_name                                             1.703561
activity_index                                            1.115531
gross_income_of_the_household                            17.841475
saving_account                                            0.000000
guarantees                                                0.000000
current_account                                           0.000000
derivative_account                                        0.000000
payroll_account                                           0.000000
junior_account                                            0.000000
mas_particular_account                                    0.000000
particular_account                                        0.000000
particular_plus_account                                   0.000000
short_term_deposits                                       0.000000
medium_term_deposits                                      0.000000
long_term_deposits                                        0.000000
e-account                                                 0.000000
funds                                                     0.000000
mortgage                                                  0.000000
pensions                                                  0.000000
loans                                                     0.000000
taxes                                                     0.000000
credit_card                                               0.000000
securities                                                0.000000
home_account                                              0.000000
payroll                                                   0.574934
pensions                                                  0.574934
direct_debit                                              0.000000
dtype: float64
```

Fig 3: Percentage of Missing values across all features.

A couple of columns have approximately 99.9 % missing data, followed by a third feature with around 17% missing data. There are other columns with much fewer missing data (less than 1%).

2. Duplicated Observations

```
cust_seg_df["customer_code"].value_counts()
```

```
87264      2
570366     2
467398     2
560121     2
194445     2
          ..
1323637    1
1325684    1
1313394    1
1315441    1
1050623    1
Name: customer_code, Length: 626159, dtype: int64
```

```
cust_seg_df = cust_seg_df.drop_duplicates(subset = ["customer_code"], keep = "last").reset_index(drop = True)
print(cust_seg_df.shape)
```

```
(626159, 47)
```

Fig 4: Count of unique Customer codes

The above picture represents a count of unique customer codes. Out of a million observations, there are only approximately 626K unique customer codes. This means that there are around 370k duplicate observations, where there are 2 customer codes.

3. High Cardinality / Disproportionate ratio of categories

```
sns.countplot(cust_seg_df["employee_index"]);
```

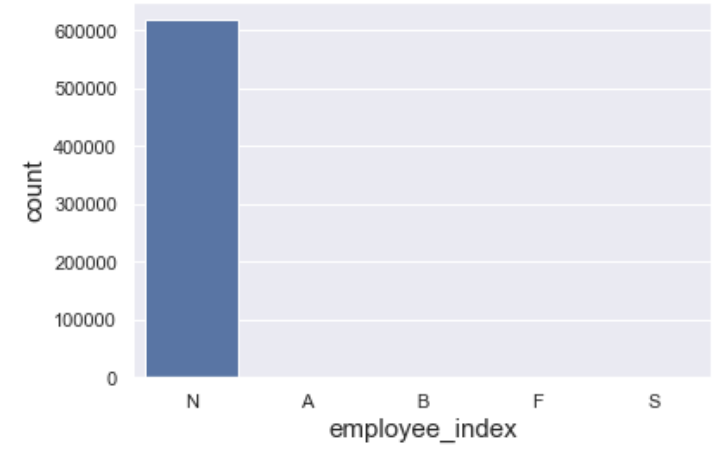

Fig 5: Countplot illustrating value counts of each category for `employee_index` feature

The above figure illustrates the value counts of each category for a chosen feature. The counts of some categories are negligible. Moreover, some features have a very high number of categories.

4. Outliers/Skewed Features

```
fig, ax = plt.subplots(1,2, figsize = (13,5));

sns.histplot(cust_seg_df["gross_income_of_the_household"], kde = True, ax = ax[0]);
sns.boxplot(x = "gross_income_of_the_household", data = cust_seg_df, ax = ax[1]);


plt.suptitle("Income Distribution of Customers");
```

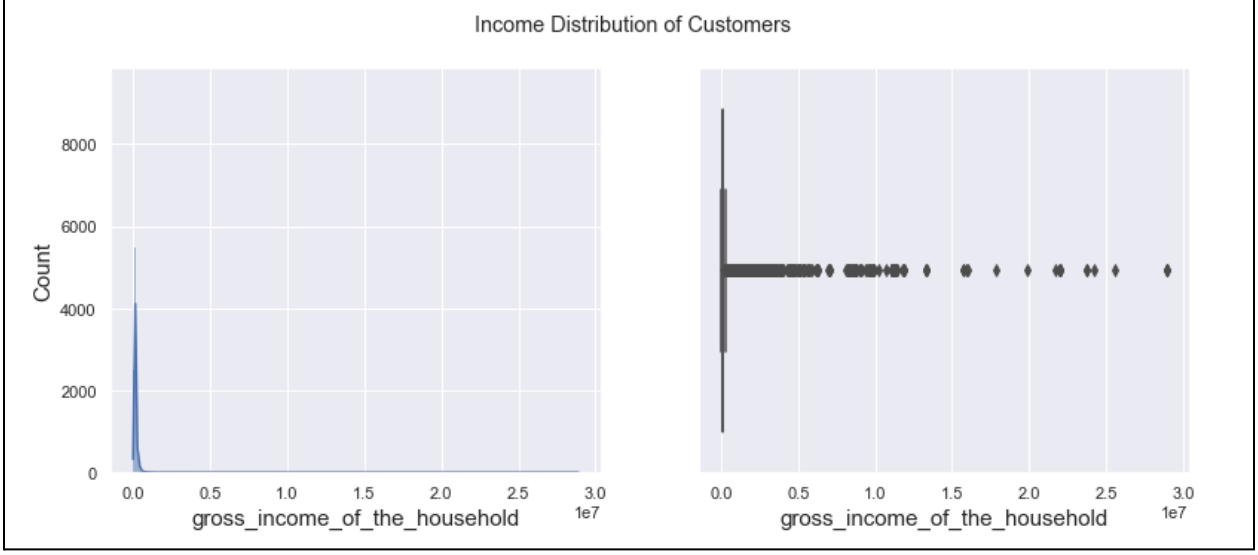

Fig 6: Distribution of Customer household income

Some of the features are extremely skewed, as shown above. There are significant outliers. This can affect the clustering model. Hence, appropriate outlier engineering techniques must be utilized.

What approaches are you trying to apply on your data set to overcome problems like NA value, outlier etc and why?

All of these approaches below ensure to preserve most of the useful observations within the dataset.

1. **Duplicated Observations**

   Most of the customer codes are repeated one time. We only need unique customer codes. Therefore, we will drop duplicated customer code observations and keep only the last observation unique for each customer code. We keep the last observation unique as the last observation for each customer code seems to be the latest entry.

2. **Missing Values**

   We will drop those features column-wise having more than 20% missing values. For the rest of the observations having missing values, we will consider them as Missing at Random, and try to impute those missing values using the information from other features.

   For example, for the customer income feature (having around 17% missing values), we will impute this feature by aggregating the province name and taking the median household income for each province. Those observations having several missing values across most of the other columns are dropped row-wise as there is no useful information that can be used from other features.

   For most categorical features with missing values, we will impute them with the mode category of that particular column. But for the province name column, which seems to have no name for a particular column, we will fill in the missing value for these observations as “Foreign”. We do this, because we noticed the missing values are for those observations that do not have a customer country residence category as ‘ES’ (Spain).

3. **Outlier**

   For the Age variable, we will replace ages below 20 with the mean of ages between 20 and 35. For ages above 85, we will replace ages with mean of ages between 35 and 85.

   For the customer house income variable, we will apply BoxCox transformation, in order to transform its distribution to nearly normal distribution. The resulting outliers will then be winsorized at both ends. Winsorizing will help distribute the outliers at both ends closer to other values within the normal distribution.