Customer Segmentation Project

Report date: 19-March-2022

Internship Batch: LISUM06

Specialization: Data Science

GitHub link: https://github.com/saadbinmunir/Customer-Segmentation-Project

Team member details:

Saad Bin Munir

- Email: saadmunir24@gmail.com
- Country: United Kingdom
- University: University of Central Lancashire

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 personalised offer to a particular group of customers. It is not effective to manually start understanding the category of the customer because they will not be able to uncover the hidden pattern in data. ABC analytics assigned this talk to their analytics team and instructed their team to come up with the approach and feature which group similar behavior customer in one category and others in different category.

Data understanding

The dataset consists of details of Bank customers from 1995 to 2015. The observation in the dataset correspond to unique customer in the dataset. The attributes contain information of each customer such as gender, location, joining date, residence and the products utilised. An overview of the dataset can be seen below.

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 type

The dataset provided contains 1000000 rows and 48 columns and is provided in csv format. It consists of categorical, numerical values as well as datatime format. The size of the dataset is 366 MB.

Most of the categorical values are binary however there are some columns with multiple categorical types. All the numerical columns contains integers and the ‘renta’ column consists of continuous values. Some of the values of the categorical columns contain float values. The complete details about the datatypes of each column as well as the overview of the dataset can been visualised in the figure 1 and figure 2 respectively.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000000 entries, 0 to 999999
Data columns (total 48 columns):
 #   Column                 Non-Null Count    Dtype
---  ------                 --------------    -----
 0   Unnamed: 0             1000000 non-null  int64
 1   fecha_dato             1000000 non-null  object
 2   ncodpers               1000000 non-null  int64
 3   ind_empleado           989218 non-null   object
 4   pais_residencia        989218 non-null   object
 5   sexo                   989214 non-null   object
 6   age                    1000000 non-null  object
 7   fecha_alta             989218 non-null   object
 8   ind_nuevo              989218 non-null   float64
 9   antiguedad             1000000 non-null  object
 10  indrel                 989218 non-null   float64
 11  ult_fec_cli_1t         1101 non-null     object
 12  indrel_1mes            989218 non-null   float64
 13  tiprel_1mes            989218 non-null   object
 14  indresi                989218 non-null   object
 15  indext                 989218 non-null   object
 16  conyuemp               178 non-null      object
 17  canal_entrada          989139 non-null   object
 18  indfall                989218 non-null   object
 19  tipodom                989218 non-null   float64
 20  cod_prov               982266 non-null   float64

 21  nomprov                982266 non-null   object
 22  ind_actividad_cliente  989218 non-null   float64
 23  renta                  824817 non-null   float64
```

s

```
21  nomprov                  982266 non-null   object
22  ind_actividad_cliente    989218 non-null   float64
23  renta                    824817 non-null   float64
24  ind_ahor_fin_ult1        1000000 non-null  int64
25  ind_aval_fin_ult1        1000000 non-null  int64
26  ind_cco_fin_ult1         1000000 non-null  int64
27  ind_cder_fin_ult1        1000000 non-null  int64
28  ind_cno_fin_ult1         1000000 non-null  int64
29  ind_ctju_fin_ult1        1000000 non-null  int64
30  ind_ctma_fin_ult1        1000000 non-null  int64
31  ind_ctop_fin_ult1        1000000 non-null  int64
32  ind_ctpp_fin_ult1        1000000 non-null  int64
33  ind_deco_fin_ult1        1000000 non-null  int64
34  ind_deme_fin_ult1        1000000 non-null  int64
35  ind_dela_fin_ult1        1000000 non-null  int64
36  ind_ecue_fin_ult1        1000000 non-null  int64
37  ind_fond_fin_ult1        1000000 non-null  int64
38  ind_hip_fin_ult1         1000000 non-null  int64
39  ind_plan_fin_ult1        1000000 non-null  int64
40  ind_pres_fin_ult1        1000000 non-null  int64
41  ind_reca_fin_ult1        1000000 non-null  int64
42  ind_tjcr_fin_ult1        1000000 non-null  int64
43  ind_valo_fin_ult1        1000000 non-null  int64
44  ind_viv_fin_ult1         1000000 non-null  int64
45  ind_nomina_ult1          994598 non-null   float64
46  ind_nom_pens_ult1        994598 non-null   float64
47  ind_recibo_ult1          1000000 non-null  int64
dtypes: float64(9), int64(24), object(15)
memory usage: 366.2+ MB
```

Figure 1. Full detail of columns

```
df = df.drop(df.columns[0], axis = 1)
df.head()
```

	fecha_dato	ncodpers	ind_empleado	pais_residencia	sexo	age	fecha_alta	ind_nuevo	antiguedad	i
0	2015-01-28	1375586	N	ES	H	35	2015-01-12	0.0	6	
1	2015-01-28	1050611	N	ES	V	23	2012-08-10	0.0	35	
2	2015-01-28	1050612	N	ES	V	23	2012-08-10	0.0	35	
3	2015-01-28	1050613	N	ES	H	22	2012-08-10	0.0	35	
4	2015-01-28	1050614	N	ES	V	23	2012-08-10	0.0	35	

5 rows × 47 columns

Figure 2. Overview of the dataset

Data Problems

The dataset is quite messy and contains a lot of missing and duplicated values. Moreover, the data is highly unbalanced which can affect the training of the model. The problem associated with the data can be seen in the figures below.

1. Missing Data

A complete overview of missing data for all columns is shown in figure 3. It can be seen that there are some columns which have a huge amount of missing data.

```
df.isna().sum().sort_values(ascending = False)
```

```
conyuemp                 999822
ult_fec_cli_1t           998899
renta                    175183
nomprov                   17734
cod_prov                  17734
canal_entrada             10861
sexo                      10786
tiprel_1mes               10782
indrel_1mes               10782
indfall                   10782
indext                    10782
indresi                   10782
ind_empleado              10782
ind_actividad_cliente     10782
indrel                    10782
ind_nuevo                 10782
fecha_alta                10782
pais_residencia           10782
tipodom                   10782
ind_nomina_ult1            5402
ind_nom_pens_ult1          5402
ncodpers                      0
antiguedad                    0
age                           0
```

Figure 3. Information about missing values

2. Duplicate Data

It is interesting to see in figure 4 that there are a lot of duplicate values in the dataset. The data contains 1 million rows however only 626k of them are unique. This shows that 374k observations have been repeated in the dataset.

```
df['ncodpers'].value_counts()
```

```
1243926    2
324414     2
296141     2
308427     2
306376     2
          ..
931530     1
933579     1
919244     1
921293     1
1048576    1
Name: ncodpers, Length: 626159, dtype: int64
```

Figure 4. Information about duplicate values

3. Unbalanced data

Figure 5 shows that the data is highly imbalanced. Some of the categories only have negligible data whereas some of them Have high number of observations.

```
sns.countplot(df['ind_empleado'])
plt.show()
```

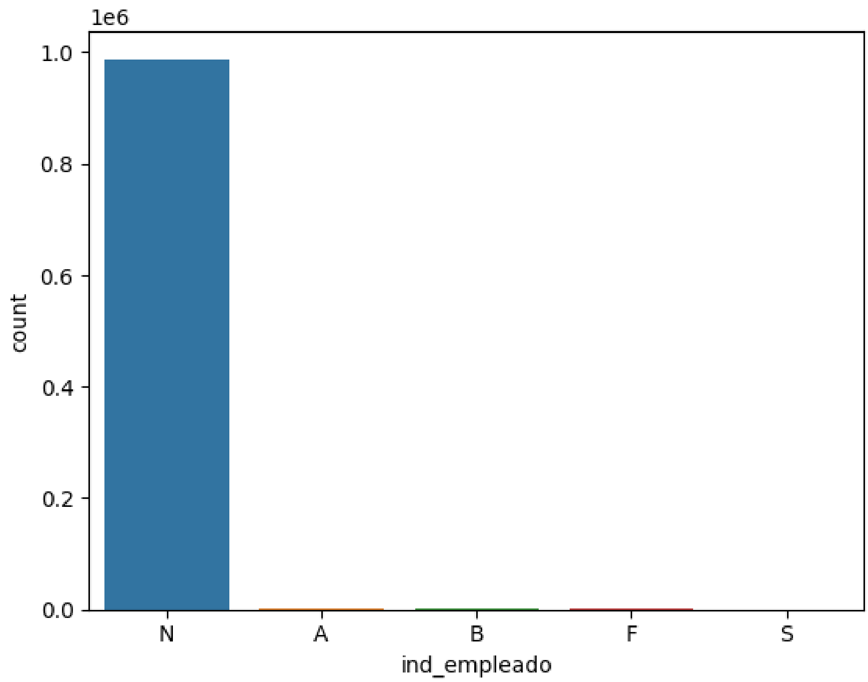

Figure 5. Information about distribution of data

4. Outliers

The figure 6 below shows the distribution of gross income of a household we can see that there are so many outliers present in the data that can affect the machine learning model hence the removal of these outliers is necessary for better results.

```
sns.boxplot(x = 'renta',data = df)
plt.show()
```

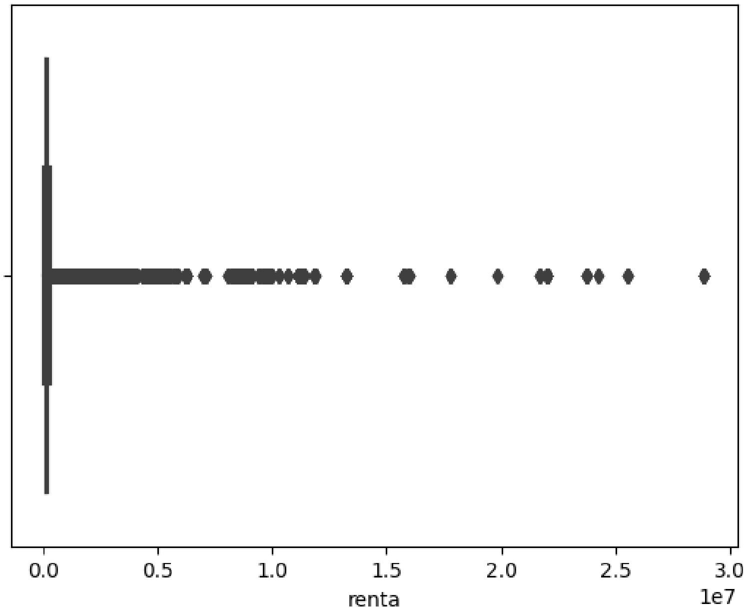

Figure 6. Information about outliers

Approach Used

1. Duplicated Observations

Since we have a lot of rows which have duplicate data hence, we will drop the duplicate values to make the data set unique. we will make sure that we drop the values which were taken earlier and retain the values which are taken later in the data set because they represent the latest data.

2. Missing Values

To deal with the missing values we dropped the data which has more than 25% of missing values because it is not efficient to impute them. The values which have small amount of missing data shall be imputed using information of other features.

For columns which have small amount of missing data set we will perform imputation. The method of imputation depends on what variable we are performing the imputation for example find the imputation of household income we will make use of median imputation. Moreover, for categorical values we will make use of model imputation.

3. Outlier

The outliers which are unrealistic can be removed as well as can be replaced by values. There are several techniques to replace the outliers for example in case of age we will use mean values to replace the outliers whereas in case of household income we will perform normal distribution which will help us to distribute the outliers.