8

April 25, 2023

1 Group Name: Go Bear!

2 Name: Xiaoke Song

3 Email: xiaokesong57@gmail.com

4 Country: born in China, college in the US

5 College: UC Berkeley

6 Specialization: Data Science

[1]:

```
import pandas as pd
```

[2]:

```
df = pd.read_csv("cust_seg.csv")
df.head(5)
```

```
/var/folders/_0/nmpfpzw134n12j0c0z6jtrw80000gn/T/ipykernel_5258/1520097819.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")
```

[2]:

```
   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]
```

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.”

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.

What type of data you have got for analysis? —— Based on the provided information, we have a dataset containing customer information for XYZ bank. The dataset includes demographic information, customer behavior, and product ownership, and it appears to be in a structured format. The warning message “Columns (16) have mixed types” indicates that there might be some columns with mixed data types, but without further information, it’s unclear which columns are affected and what the specific data types are. Overall, it seems like we have tabular data that can be analyzed using various techniques such as descriptive statistics, data visualization, and clustering algorithms for customer segmentation.

[4]:
```
# check for missing values
print('Number of missing values in each column:')
print(df.isnull().sum())
```

```
Number of missing values in each column:
Unnamed: 0                   0
fecha_dato                   0
ncodpers                     0
ind_empleado             10782
pais_residencia          10782
sexo                     10786
```

```
age                           0
fecha_alta                10782
ind_nuevo                 10782
antiguedad                    0
indrel                    10782
ult_fec_cli_1t           998899
indrel_1mes               10782
tiprel_1mes               10782
indresi                   10782
indext                    10782
conyuemp                 999822
canal_entrada             10861
indfall                   10782
tipodom                   10782
cod_prov                  17734
nomprov                   17734
ind_actividad_cliente     10782
renta                    175183
ind_ahor_fin_ult1             0
ind_aval_fin_ult1             0
ind_cco_fin_ult1              0
ind_cder_fin_ult1             0
ind_cno_fin_ult1              0
ind_ctju_fin_ult1             0
ind_ctma_fin_ult1             0
ind_ctop_fin_ult1             0
ind_ctpp_fin_ult1             0
ind_deco_fin_ult1             0
ind_deme_fin_ult1             0
ind_dela_fin_ult1             0
ind_ecue_fin_ult1             0
ind_fond_fin_ult1             0
ind_hip_fin_ult1              0
ind_plan_fin_ult1             0
ind_pres_fin_ult1             0
ind_reca_fin_ult1             0
ind_tjcr_fin_ult1             0
ind_valo_fin_ult1             0
ind_viv_fin_ult1              0
ind_nomina_ult1            5402
ind_nom_pens_ult1          5402
ind_recibo_ult1               0
dtype: int64
```

There are missing values in the following columns:

ind_empleado,

pais_residencia,

sexo,

fecha_alta,

ind_nuevo,

indrel,

ult_fec_cli_1t,

indrel_1mes,

tiprel_1mes,

indresi,

indext,

conyuemp,

canal_entrada,

indfall,

tipodom,

cod_prov,

nomprov,

ind_actividad_cliente,

renta,

ind_nomina_ult1,

ind_nom_pens_ult1;

The column "ult_fec_cli_1t" has a very large number of missing values, while the columns "ind_empleado", "pais_residencia", "sexo", "fecha_alta", "ind_nuevo", "indrel", "indrel_1mes", "tiprel_1mes", "indresi", "indext", "conyuemp", "canal_entrada", "indfall", "tipodom", "cod_prov", "nomprov", "ind_actividad_cliente", "renta", "ind_nomina_ult1", and "ind_nom_pens_ult1" have a relatively smaller number of missing values.

[3]:

```
# calculate the summary statistics
summary = df.describe()

# print the summary statistics
print(summary)
```

```
           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]
```

The 75th percentile refers to the point in a dataset where 75% of the observations are below that value. Outliers are observations that fall outside of the typical range of values for a dataset, and can be identified using various statistical methods. Common methods for identifying outliers include the use of box plots, z-scores, and interquartile range (IQR).

[4]:

```python
# calculate the skewness
skewness = df.skew()

# print the skewness
print(skewness)
```

```
Unnamed: 0              -2.511790e-15
ncodpers                 3.227844e-02
ind_nuevo                4.517572e+01
indrel                   2.992451e+01
indrel_1mes              1.534514e+02
tipodom                  0.000000e+00
cod_prov                -1.512119e-01
ind_actividad_cliente   -2.621046e-01
renta                    5.223413e+01
ind_ahor_fin_ult1        7.514476e+01
ind_aval_fin_ult1        1.601190e+02
ind_cco_fin_ult1        -1.152401e+00
ind_cder_fin_ult1        4.109809e+01
ind_cno_fin_ult1         2.571915e+00
ind_ctju_fin_ult1        8.391620e+00
ind_ctma_fin_ult1        9.903618e+00
ind_ctop_fin_ult1        1.405709e+00
ind_ctpp_fin_ult1        3.309287e+00
ind_deco_fin_ult1        2.145683e+01
ind_deme_fin_ult1        1.773314e+01
ind_dela_fin_ult1        3.467512e+00
ind_ecue_fin_ult1        2.555226e+00
ind_fond_fin_ult1        5.815247e+00
ind_hip_fin_ult1         9.858534e+00
ind_plan_fin_ult1        8.107362e+00
ind_pres_fin_ult1        1.454481e+01
ind_reca_fin_ult1        3.294845e+00
ind_tjcr_fin_ult1        3.493287e+00
ind_valo_fin_ult1        4.736660e+00
ind_viv_fin_ult1         1.233849e+01
```

```
ind_nomina_ult1          3.322353e+00
ind_nom_pens_ult1        3.107782e+00
ind_recibo_ult1          1.792646e+00
dtype: float64
```

```
/var/folders/_0/nmpfpzw134n12j0c0z6jtrw80000gn/T/ipykernel_5258/2832739648.py:2:
FutureWarning: Dropping of nuisance columns in DataFrame reductions (with
'numeric_only=None') is deprecated; in a future version this will raise
TypeError.  Select only valid columns before calling the reduction.
  skewness = df.skew()
```

To determine the skewness of the data, we need to look at the signs of the values in the third column. If the value is positive, the distribution is right-skewed (or positively skewed), and if the value is negative, the distribution is left-skewed (or negatively skewed).

From the given data, we can see that:

Unnamed: 0, ncodpers, tipodom, ind_cco_fin_ult1, ind_cno_fin_ult1, ind_ctop_fin_ult1, ind_ctpp_fin_ult1,

ind_dela_fin_ult1, ind_ecue_fin_ult1, ind_reca_fin_ult1, ind_tjcr_fin_ult1, ind_valo_fin_ult1, ind_nomina_ult1,

ind_nom_pens_ult1, and ind_recibo_ult1 have a skewness value of approximately zero (around 0).

ind_nuevo, indrel, ind_cder_fin_ult1, ind_ctju_fin_ult1, ind_ctma_fin_ult1, ind_deco_fin_ult1, ind_deme_fin_ult1,

ind_fond_fin_ult1, ind_hip_fin_ult1, ind_plan_fin_ult1, ind_pres_fin_ult1, and ind_viv_fin_ult1 have positive skewness values, indicating a right-skewed distribution. cod_prov and ind_actividad_cliente have negative skewness values, indicating a left-skewed distribution. indrel_1mes and ind_aval_fin_ult1 have very high positive skewness values, indicating extreme right-skewed distributions. Therefore, we can say that the given dataset has a mix of left-skewed, right-skewed, and extreme right-skewed distributions.

What approaches you are trying to apply on your data set to overcome problems like NA value, outlier etc and why? —— one approach to handle missing values is imputation using the mean value of the column. Here's an example code below. In this code, we first load the data and then check for missing values using isnull() and sum() methods. Then, we replace the missing values with the mean value of the column using fillna() method with inplace=True to replace the missing values in the original data frame. Finally, we check for missing values again to confirm that all missing values have been replaced with the mean value of the column.

[ ]:

```
import numpy as np

# check for missing values
print(df.isnull().sum())

# replace missing values with mean value of column
df.fillna(df.mean(), inplace=True)
```

```
# check for missing values again
print(df.isnull().sum())
```

```
Unnamed: 0                    0
fecha_dato                    0
ncodpers                      0
ind_empleado              10782
pais_residencia           10782
sexo                      10786
age                           0
fecha_alta                10782
ind_nuevo                 10782
antiguedad                    0
indrel                    10782
ult_fec_cli_1t           998899
indrel_1mes               10782
tiprel_1mes               10782
indresi                   10782
indext                    10782
conyuemp                 999822
canal_entrada             10861
indfall                   10782
tipodom                   10782
cod_prov                  17734
nomprov                   17734
ind_actividad_cliente     10782
renta                    175183
ind_ahor_fin_ult1             0
ind_aval_fin_ult1             0
ind_cco_fin_ult1              0
ind_cder_fin_ult1             0
ind_cno_fin_ult1              0
ind_ctju_fin_ult1             0
ind_ctma_fin_ult1             0
ind_ctop_fin_ult1             0
ind_ctpp_fin_ult1             0
ind_deco_fin_ult1             0
ind_deme_fin_ult1             0
ind_dela_fin_ult1             0
ind_ecue_fin_ult1             0
ind_fond_fin_ult1             0
ind_hip_fin_ult1              0
ind_plan_fin_ult1             0
ind_pres_fin_ult1             0
ind_reca_fin_ult1             0
ind_tjcr_fin_ult1             0
ind_valo_fin_ult1             0
```

```
ind_viv_fin_ult1           0
ind_nomina_ult1         5402
ind_nom_pens_ult1       5402
ind_recibo_ult1            0
dtype: int64
```

```
[ ]:
```