Team: Solo
Group Name:Solo Group
Email:Furkanberatay27@gmail.com
College: Hunter College

Problem description : XYZ credit union in Latin America is performing very well in selling the Banking products (eg: Credit card, deposit account, retirement account, safe deposit box etc) but their existing customer is not not buying more than 1 product which means bank is not performing good in cross selling (Bank is not able to sell their other offerings to existing customer). XYZ Credit Union decided to approach ABC analytics to solve their problem.

Data Cleansing and Transformation Done on data:

I renamed Some of the column names to have a better understanding of the columns I was working with.

Replaced the spanish words to english to be able to better understand and create correlations with the data

Then I checked each column I would be working with for some of its values and its null values I used isnull.any to check if there are any null values

```
    print( "Values: ")
    print(df['age'].value_counts())
    print("Null Values: ")
    print(df[['age']].isnull().any())
```

```
Values:
age
24     50706
23     49604
22     47674
21     46322
25     41429
       ...
114        6
117        1
164        1
118        1
127        1
Name: count, Length: 118, dtype: int64
Null Values:
age    False
dtype: bool
```

```
    print(" Values: ")
    print(df['customer_Registration'].value_counts())
    print("Null Values: ")
    print(df[['customer_Registration']].isnull().any())
```

```
 Values:
customer_Registration
2014-07-28    3421
2014-10-03    3355
2014-08-04    2787
2013-10-14    2633
2013-08-03    2013
              ...
2013-06-15       1
2012-04-29       1
2014-12-13       1
2014-04-26       1
2013-09-22       1
Name: count, Length: 6750, dtype: int64
Null Values:
customer_Registration    False
dtype: bool
```

```
    print(" Values: ")
    print(df['primary_Status'].value_counts())
    print("Null Values: ")
    print(df[['primary_Status']].isnull().any())
```

```
 Values:
primary_Status
1     927932
99      1683
Name: count, dtype: int64
Null Values:
primary_Status    False
dtype: bool
```

```
    print(" Values: ")
    print(df['Last_primary_status'].value_counts())
    print("Null Values: ")
    print(df[['Last_primary_status']].isnull().any())
```

```
 Values:
Last_primary_status
2016-06-01    138
2016-06-10    133
2016-06-03    110
2016-06-07    102
2016-06-06    101
2016-06-13     84
2016-06-20     84
2016-06-17     78
2016-06-15     78
2016-06-23     78
2016-06-14     76
2016-06-02     75
2016-06-09     75
2016-06-22     72
2016-06-21     70
2016-06-24     64
2016-06-16     62
2016-06-08     60
2016-06-27     58
2016-06-28     49
2016-06-29     36
Name: count, dtype: int64
Null Values:
Last_primary_status    True
dtype: bool
```

```
    print(" Values: ")
    print(df['gross_income'].value_counts())
    print("Null Values: ")
    print(df[['gross_income']].isnull().any())
```

```
 Values:
gross_income
       NA    227965
451931.22       354
463625.16       111
128318.52        91
181042.20        91
              ...
 41400.81         1
 47322.18         1
175518.57         1
105938.64         1
111644.01         1
Name: count, Length: 516403, dtype: int64
Null Values:
gross_income    False
dtype: bool
```

```
Null Values:
prov_name    True
dtype: bool
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

```
    print(" Values: ")
    print(df['activity_index'].value_counts())
    print("Null Values: ")
    print(df[['activity_index']].isnull().any())
```

```
 Values:
activity_index
0    534276
1    395339
Name: count, dtype: int64
Null Values:
activity_index    False
dtype: bool
```

```python
    print(" Values: ")
    print(df['prov_name'].value_counts())
    print("Null Values: ")
    print(df[['prov_name']].isnull().any())
```

```
 Values:
prov_name
MADRID                    298250
BARCELONA                  88579
VALENCIA                   47996
SEVILLA                    40492
CORUÑA, A                  28715
MURCIA                     27752
MALAGA                     24546
ZARAGOZA                   23160
ALICANTE                   22147
CADIZ                      19795
PONTEVEDRA                 18961
ASTURIAS                   18300
PALMAS, LAS                16332
VALLADOLID                 16018
BADAJOZ                    12936
TOLEDO                     12658
BIZKAIA                    12494
GRANADA                    12392
SALAMANCA                  11071
CANTABRIA                  10824
CORDOBA                     9831
BALEARS, ILLES              9130
CACERES                     8598
...
Name: count, dtype: int64
```

```python
    print(" Values: ")
    print(df['type_of_owner'].value_counts())
    print("Null Values: ")
    print(df[['type_of_owner']].isnull().any())
```

```
 Values:
type_of_owner
1.0    929565
3.0        27
Name: count, dtype: int64
Null Values:
type_of_owner    True
dtype: bool
```

```python
    print(" Values: ")
    print(df['prov_name'].value_counts())
    print("Null Values: ")
    print(df[['prov_name']].isnull().any())
```

```
 Values:
prov_name
MADRID                    298250
BARCELONA                  88579
VALENCIA                   47996
SEVILLA                    40492
CORUÑA, A                  28715
MURCIA                     27752
MALAGA                     24546
ZARAGOZA                   23160
ALICANTE                   22147
CADIZ                      19795
PONTEVEDRA                 18961
ASTURIAS                   18300
PALMAS, LAS                16332
VALLADOLID                 16018
BADAJOZ                    12936
TOLEDO                     12658
BIZKAIA                    12494
GRANADA                    12392
SALAMANCA                  11071
CANTABRIA                  10824
CORDOBA                     9831
BALEARS, ILLES              9130
CACERES                     8598
...
Name: count, dtype: int64
```

```
    print(" Values: ")
    print(df['segmentation'].value_counts())
    print("Null Values: ")
    print(df[['segmentation']].isnull().any())
]
```

```
 Values:
segmentation
02 - PARTICULARES     545378
03 - UNIVERSITARIO    346028
01 - TOP               35961
Name: count, dtype: int64
Null Values:
segmentation    True
dtype: bool
```

```
    print(" Values: ")
    print(df['relation_type'].value_counts())
    print("Null Values: ")
    print(df[['relation_type']].isnull().any())
```

```
 Values:
relation_type
I    535943
A    393622
P        27
Name: count, dtype: int64
Null Values:
relation_type    True
dtype: bool
```

```
    print(" Values: ")
    print(df['channel_used'].value_counts())
    print("Null Values: ")
    print(df[['channel_used']].isnull().any())
```

```
 Values:
channel_used
KHE    251665
KAT    205833
KFC    200697
KHQ     74969
KHM     33384
```

```
KDB         1
KHR         1
KGN         1
025         1
KDL         1
Name: count, Length: 162, dtype: int64
Null Values:
channel_used    True
dtype: bool
```

```
print(" Values: ")
print(df['deceased_index'].value_counts())
print("Null Values: ")
print(df[['deceased_index']].isnull().any())
```

[19]

```
 Values:
deceased_index
No     927215
Yes      2400
Name: count, dtype: int64
Null Values:
deceased_index    False
dtype: bool
```

```
print(" Values: ")
print(df['primary_adrss'].value_counts())
print("Null Values: ")
print(df[['primary_adrss']].isnull().any())
```

[20]

```
 Values:
primary_adrss
1    929615
Name: count, dtype: int64
Null Values:
primary_adrss    False
dtype: bool
```

```
    print(" Values: ")
    print(df['prov_code'].value_counts())
    print("Null Values: ")
    print(df[['prov_code']].isnull().any())
```

```
 Values:
prov_code
28.0    298250
8.0      88579
46.0     47996
41.0     40492
15.0     28715
30.0     27752
29.0     24546
50.0     23160
3.0      22147
11.0     19795
36.0     18961
33.0     18300
35.0     16332
47.0     16018
```

```
    print(" Values: ")
    print(df['deceased_index'].value_counts())
    print("Null Values: ")
    print(df[['deceased_index']].isnull().any())
```

22]

```
 Values:
deceased_index
No     927215
Yes      2400
Name: count, dtype: int64
Null Values:
deceased_index    False
dtype: bool
```

```
    print("Values:  ")
    print(df['data_Partition'].value_counts())
    print("Null Values: ")
    print(df[['data_Partition']].isnull().any())
```

```
Values:
data_Partition
2016-06-28    929615
Name: count, dtype: int64
Null Values:
data_Partition    False
dtype: bool
```

```
    print("Values: ")
    print(df['employee_Index'].value_counts())
    print("Null Values: ")
    print(df[['employee_Index']].isnull().any())
```

```
Values:
employee_Index
N    929096
B       218
F       152
A       148
S         1
Name: count, dtype: int64
Null Values:
employee_Index    False
dtype: bool
```

After checking the null values I filled the values up using .ffill() method and bfill() for the first value of the table

```
df['Last_primary_status'] = df['Last_primary_status'].ffill()
df['Last_primary_status'] = df['Last_primary_status'].bfill()
df['type_of_owner'] = df['type_of_owner'].ffill()
df['relation_type'] = df['relation_type'].ffill()
df['channel_used'] = df['channel_used'].ffill()
df['prov_code'] = df['prov_code'].ffill()
df['prov_name'] = df['prov_name'].ffill()
df['segmentation'] = df['segmentation'].ffill()
```

Afterwards I checked if there were any null values left using isnull().sum() which shows the sum of all null values in a column

```
df.isnull().sum()
```

```
data_Partition            0
employee_Index            0
age                       0
customer_Registration     0
primary_Status            0
Last_primary_status       0
type_of_owner             0
relation_type             0
channel_used              0
deceased_index            0
primary_adrss             0
prov_code                 0
prov_name                 0
activity_index            0
gross_income              0
segmentation              0
dtype: int64
```

Finally I made a boxplot to look for outliers

```
plt.figure(figsize=(20,15))
ax=sns.boxplot(data = df, palette='BuPu')
plt.xticks(rotation=90)

plt.show()
```