Data Analyst: Cross selling recommendation

1. Team member's details

Group Name: Dreamers

Name: Surabhi mahawar
Email: surabhimahawar@gmail.com
Country: India
College/Company: APJ Kalam Technical University
Specialization (Data Science, NLP, Data Analyst): Data analyst

Name: Deborah Adeyemi
Email: adeyemianuoluwapod@gmail.com ,
Country: United Kingdom,
College/Company: University of the West of England(UWE)
Specialization (Data Science, NLP, Data Analyst): Data analyst

Name: Kseniia Nosenko
Email: kseniianosenko@gmail.com
Country: Germany
College/Company: Higher School of Economics
Specialization (Data Science, NLP, Data Analyst): Data analyst

Name: Pranav Walia
Email: waliap@miamioh.edu
Country: America
College/Company: Miami University
Specialization (Data Science, NLP, Data Analyst): Data analyst

2. Problem description

XYZ Credit union in Latin America is performing very well in selling Banking products (Credit card , deposit amount, retirement account, safe deposit box), but their existing customer is not buying more than 1 product which means bank is not performing good in cross selling (Banks is not able to sell their other offering to existing customers). XYZ credit union decided to approach ABC analytics to solve their problem.

3. Data understanding

There are 2 files: Test and Train. Test file hasn't any info about products and we can’t use this file for analysis.

Let’s check the number of rows in train datasets, what type of variables are there and whether values are null (info in %).

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13647309 entries, 0 to 13647308
Data columns (total 48 columns):
 #   Column                 Dtype
---  ------                 -----
 0   fecha_dato             object
 1   ncodpers               int64
 2   ind_empleado           object
 3   pais_residencia        object
 4   sexo                   object
 5   age                    object
 6   fecha_alta             object
 7   ind_nuevo              float64
 8   antiguedad             object
 9   indrel                 float64
 10  ult_fec_cli_1t         object
 11  indrel_1mes            object
 12  tiprel_1mes            object
 13  indresi                object
 14  indext                 object
 15  conyuemp               object
 16  canal_entrada          object
 17  indfall                object
 18  tipodom                float64
 19  cod_prov               float64
 20  nomprov                object
 21  ind_actividad_cliente  float64
 22  renta                  float64
 23  segmento               object
 24  ind_ahor_fin_ult1      int64
 25  ind_aval_fin_ult1      int64
 26  ind_cco_fin_ult1       int64
 27  ind_cder_fin_ult1      int64
 28  ind_cno_fin_ult1       int64
 29  ind_ctju_fin_ult1      int64
 30  ind_ctma_fin_ult1      int64
 31  ind_ctop_fin_ult1      int64
 32  ind_ctpp_fin_ult1      int64
 33  ind_deco_fin_ult1      int64
 34  ind_deme_fin_ult1      int64
 35  ind_dela_fin_ult1      int64
 36  ind_ecue_fin_ult1      int64
 37  ind_fond_fin_ult1      int64
 38  ind_hip_fin_ult1       int64
 39  ind_plan_fin_ult1      int64
 40  ind_pres_fin_ult1      int64
 41  ind_reca_fin_ult1      int64
 42  ind_tjcr_fin_ult1      int64
 43  ind_valo_fin_ult1      int64
 44  ind_viv_fin_ult1       int64
 45  ind_nomina_ult1        float64
 46  ind_nom_pens_ult1      float64
 47  ind_recibo_ult1        int64
dtypes: float64(8), int64(23), object(17)
memory usage: 4.9+ GB
```

```
fecha_dato                0.000000
ncodpers                  0.000000
ind_empleado              0.203220
pais_residencia           0.203220
sexo                      0.203732
age                       0.000000
fecha_alta                0.203220
ind_nuevo                 0.203220
antiguedad                0.000000
indrel                    0.203220
ult_fec_cli_1t           99.818330
indrel_1mes               1.097513
tiprel_1mes               1.097513
indresi                   0.203220
indext                    0.203220
conyuemp                 99.986752
canal_entrada             1.363829
indfall                   0.203220
tipodom                   0.203227
cod_prov                  0.685784
nomprov                   0.685784
ind_actividad_cliente     0.203220
renta                    20.475648
segmento                  1.387585
ind_ahor_fin_ult1         0.000000
ind_aval_fin_ult1         0.000000
ind_cco_fin_ult1          0.000000
ind_cder_fin_ult1         0.000000
ind_cno_fin_ult1          0.000000
ind_ctju_fin_ult1         0.000000
ind_ctma_fin_ult1         0.000000
ind_ctop_fin_ult1         0.000000
ind_ctpp_fin_ult1         0.000000
ind_deco_fin_ult1         0.000000
ind_deme_fin_ult1         0.000000
ind_dela_fin_ult1         0.000000
ind_ecue_fin_ult1         0.000000
ind_fond_fin_ult1         0.000000
ind_hip_fin_ult1          0.000000
ind_plan_fin_ult1         0.000000
ind_pres_fin_ult1         0.000000
ind_reca_fin_ult1         0.000000
ind_tjcr_fin_ult1         0.000000
ind_valo_fin_ult1         0.000000
ind_viv_fin_ult1          0.000000
ind_nomina_ult1           0.117701
ind_nom_pens_ult1         0.117701
ind_recibo_ult1           0.000000
dtype: float64
```

Train dataset contains info about customers and their products.
Train dataset contains 48 features and 13 647 309 rows. There are no duplicates.

Some of features was defined as object, but it is Numerical Variables: `age`, `antiguedad`.
Some features in opposite were defined as float64, but it is categorical features:
`ind_nuevo`, `indrel`, `tipodom`, `cod_prov`, `ind_actividad_cliente`

In general:
Continuous Variables: `'age'`, `'antiguedad'`, `'renta'`
Categorical Variables: others

There are some features with the same number of missing values, I expect those relate to the same rows. We delete rows with a lot of missing values (0.2% of data).

Null values after deleting rows with a lot of missing values:

```
fecha_dato                 0.000000
ncodpers                   0.000000
ind_empleado               0.000000
pais_residencia            0.000000
sexo                       0.000514
age                        0.000000
fecha_alta                 0.000000
ind_nuevo                  0.000000
antiguedad                 0.000000
indrel                     0.000000
ult_fec_cli_1t            99.817961
indrel_1mes                0.896115
tiprel_1mes                0.896115
indresi                    0.000000
indext                     0.000000
conyuemp                  99.986725
canal_entrada              1.162973
indfall                    0.000000
tipodom                    0.000007
cod_prov                   0.483547
nomprov                    0.483547
ind_actividad_cliente      0.000000
renta                     20.313710
segmento                   1.186777
ind_ahor_fin_ult1          0.000000
ind_aval_fin_ult1          0.000000
ind_cco_fin_ult1           0.000000
ind_cder_fin_ult1          0.000000
ind_cno_fin_ult1           0.000000
ind_ctju_fin_ult1          0.000000
ind_ctma_fin_ult1          0.000000
ind_ctop_fin_ult1          0.000000
ind_ctpp_fin_ult1          0.000000
ind_deco_fin_ult1          0.000000
ind_deme_fin_ult1          0.000000
ind_dela_fin_ult1          0.000000
ind_ecue_fin_ult1          0.000000
ind_fond_fin_ult1          0.000000
ind_hip_fin_ult1           0.000000
ind_plan_fin_ult1          0.000000
ind_pres_fin_ult1          0.000000
ind_reca_fin_ult1          0.000000
ind_tjcr_fin_ult1          0.000000
ind_valo_fin_ult1          0.000000
ind_viv_fin_ult1           0.000000
ind_nomina_ult1            0.001593
ind_nom_pens_ult1          0.001593
ind_recibo_ult1            0.000000
dtype: float64
```

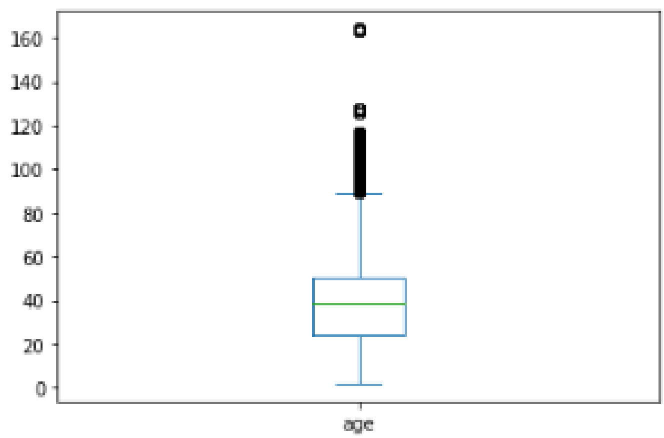
Describe data:

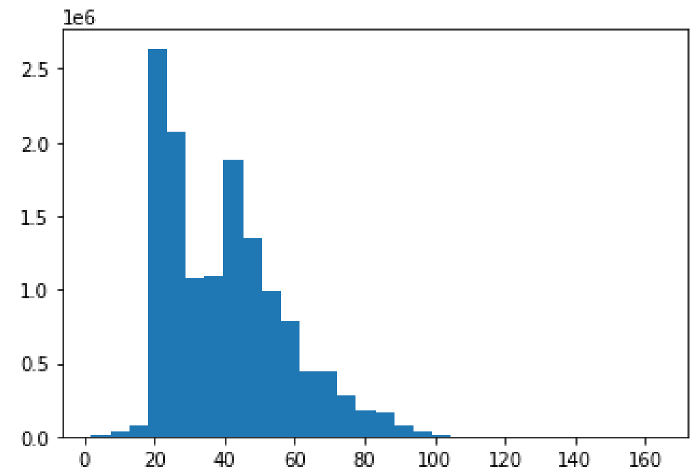
- **numeric**

	age	antiguedad	renta
count	13619575	13619575	10852934
mean	40	77	134254
std	17	1672	230620
min	2	-999999	1203
25%	24	23	68711
50%	39	50	101850
75%	50	135	155956
max	164	256	28894396

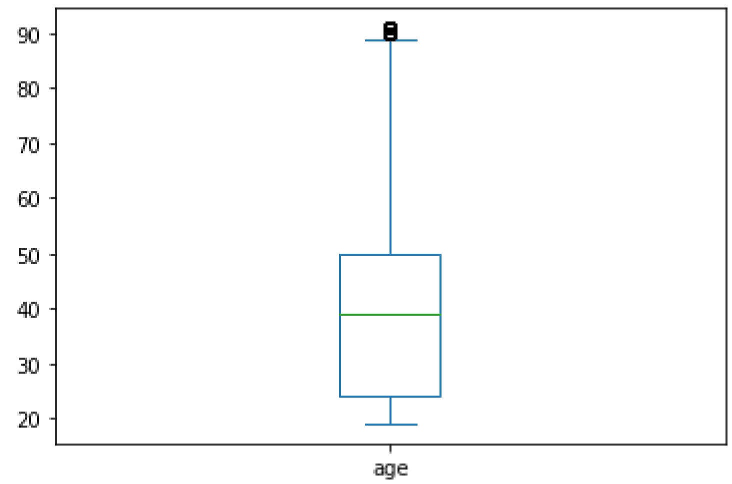
Outliers in numeric data:

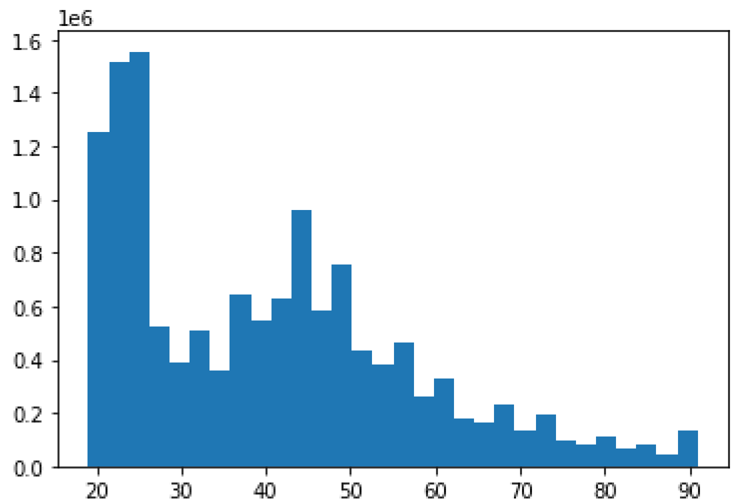
The feature `'age'` has some rows with customers who are older than 100 years and a few customers who are very young. We think there is a lot of incorrect data. We can see that customers who are older 20 and younger 90 are most. One of the ways to overcome outliers is to unite the youngest and most adult customers into groups.



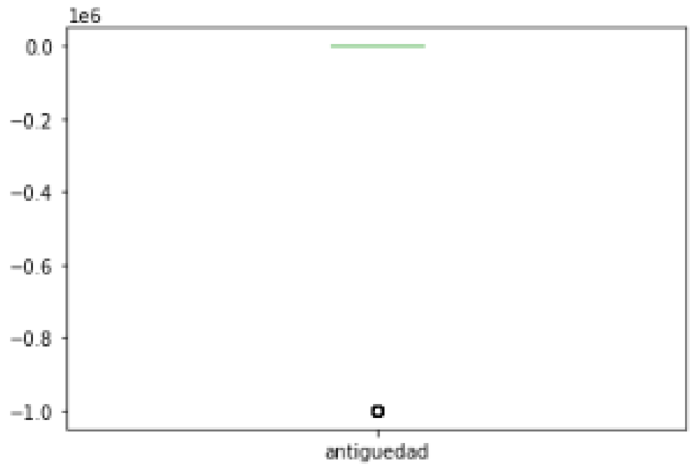


Graphs after corrections:

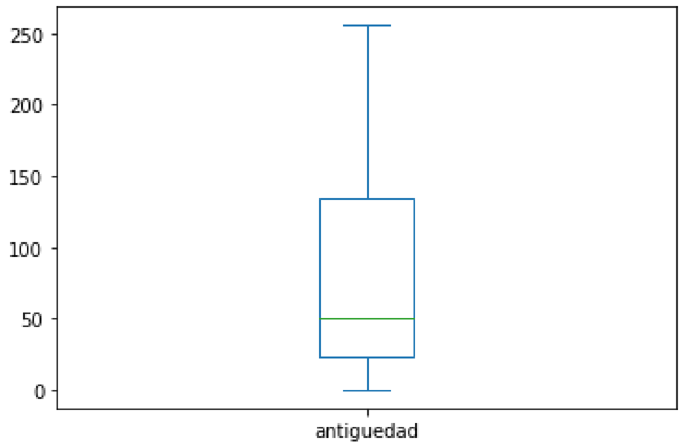




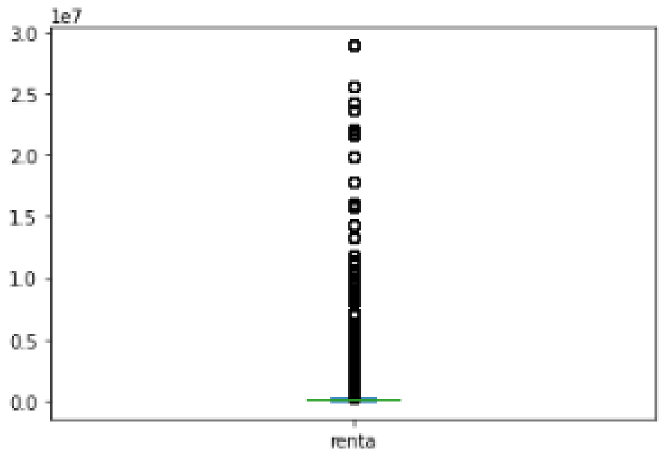
The feature `'antiguedad'` (Customer seniority (in months)) contains 38 rows with value -99999. It makes the data very skewed. We should delete rows with this value, because it is an unknown value:



The box plot feature's `'antiguedad'` after deleting:

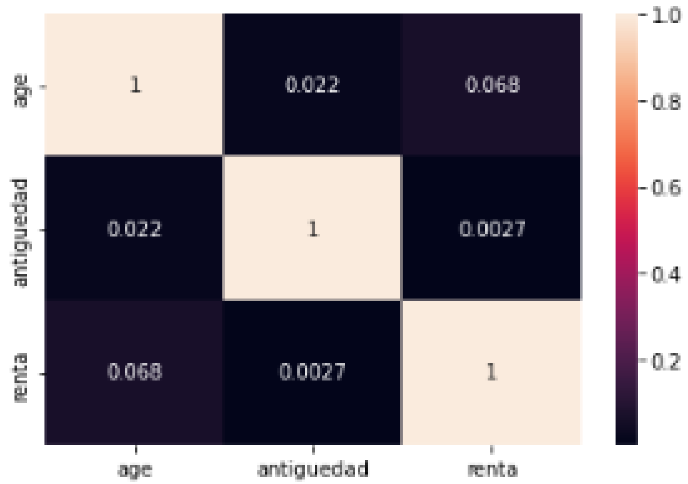


The feature `Renta` is also very shifted, because there is `18,9%` data much more than 75% quartile.



Almost 19% of data are the outliers. And 20% of data is null. Delete these rows will be incorrect. It is necessary to carry out work on the replacement of zero values and emissions. For NA values it can be for example mean/median/mode/segmented approach etc.
For outliers it can be grouping.

There is no correlation between numerical features:



- **categorical**

1. `ind_empleado` - Employee index. No NA values, 99 % rows have the value N not employee.
2. `pais_residencia` - Customer's Country residence. No NA values. 118 unique values. 99 % rows have the value ES.
3. `sexo` - Customer's sex. A small number NA values (0.0005% or 70 rows). It is necessary to carry out work on the replacement of NA values (the most popular values for example).
4. `fecha_alta` - The date in which the customer became the first holder of a contract in the bank. No NA values.
5. `ind_nuevo` - New customer Index. 1 if the customer registered in the last 6 months. Data type float, should be changed to categorical. No NA values.
6. `indrel` - 1 (First/Primary), 99 (Primary customer during the month but not at the end of the month). No NA values. If you build an ML model, it could be better to change 99 on 0 because it is scaled for ML models.
7. `ult_fec_cli_1t` - Last date as primary customer (if he isn't at the end of the month) and `conyuemp` - Spouse index. (1 if the customer is spouse of an employee). have 99% null values. According to the instructions `conyuemp` feature should contain number 1 if the customer is spouse of an employee. In dataset the feature `conyuemp` contain (N, S, nan) values. I suppose that N=No, S = Si (Yes). The number of clients with value 'S' = 17. *We can delete these features because they contain too small info for analysis.*
8. `indrel_1mes` - Customer type at the beginning of the month and `tiprel_1mes`- Customer relation type at the beginning of the month. There are 0.89% NA values. It is necessary to carry out work on the replacement of NA values.
9. `indresi` - Residence index. No NA values. 99% of customers have the same residence country as the bank country.
10. `indext` - Foreigner index. No NA values. (S (Yes) or N (No) if the customer's birth country is different than the bank country) `0.95 %` of rows have value N.

11. `canal_entrada` - channel used by the customer to join. 162 unique values. The most popular is KHE. 1.16% are NA values. We can replace the missing values with the most popular values in general or by region or something else.
12. `indfall` - Deceased index - 0.99 of rows have value N (not). No NA values.
13. `tipodom` - Addres type. 1, primary address. No NA values.
14. `cod_prov` (Province code) and `nomprov` (Province name) explain the same thing. I suppose delete feature `cod_prov` and change on categorical values feature `nomprov`. They have the same number of NA values - `0.48%.`
15. `ind_actividad_cliente` - Activity index (1, active customer; 0, inactive customer). No NA values. There are 54% inactive customers.
16. `segmento` - segmentation: 01 - VIP, 02 - Individuals 03 - college graduated. 1.18% NA values. PARTICULARES customers are 58%, UNIVERSITARIO customers are 36%, TOP customers are 4%
17. other features describe the product and customer's product availability.

Approaches to overcome problems like NA value, outlier etc:

NA values:

- replacing with a mean value where there are not many missing values, this will not affect the result due to the small volume.
- replacing the average value based on the data of another feature (Segmento-Renta).

Outliers:

- Grouping data (all clients with an age of less than 20 years, set the age of 19 years to those from 90 - 91 years)
- Moving from numeric data to categorical data (renta)