

# 01\_Exploration.ipynb

September 14, 2021

## 1 Initialisation

```
[ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.preprocessing import OrdinalEncoder, OneHotEncoder #, LabelEncoder
from sklearn.preprocessing import FunctionTransformer
from sklearn.impute import SimpleImputer
from sklearn.compose import make_column_transformer
from sklearn.pipeline import Pipeline, make_pipeline

import warnings
warnings.simplefilter(action='ignore', category=UserWarning)

from FeatureNames import get_feature_names
```

## 2 Exploration

```
[ ]: ! ls -lh ../02_data/
```

```
total 2,5G
-rw-rw-r-- 1 adrien adrien  26M juin  26  2018 application_test.csv
-rw-rw-r-- 1 adrien adrien 159M juin  26  2018 application_train.csv
-rw-rw-r-- 1 adrien adrien 359M juin  26  2018 bureau_balance.csv
-rw-rw-r-- 1 adrien adrien 163M juin  26  2018 bureau.csv
-rw-rw-r-- 1 adrien adrien 405M juin  26  2018 credit_card_balance.csv
-rw-rw-r-- 1 adrien adrien  37K juin  26  2018
HomeCredit_columns_description.csv
-rw-rw-r-- 1 adrien adrien 690M juin  26  2018 installments_payments.csv
-rw-rw-r-- 1 adrien adrien 375M juin  26  2018 POS_CASH_balance.csv
-rw-rw-r-- 1 adrien adrien 387M juin  26  2018 previous_application.csv
-rw-rw-r-- 1 adrien adrien 524K juin  26  2018 sample_submission.csv
```

```
[ ]: col_desc = pd.read_csv('../02_data/HomeCredit_columns_description.csv',
                             index_col=0)
col_desc
```

```
[ ]:
```

	Table	Row \
1	application_{train test}.csv	SK_ID_CURR
2	application_{train test}.csv	TARGET
5	application_{train test}.csv	NAME_CONTRACT_TYPE
6	application_{train test}.csv	CODE_GENDER
7	application_{train test}.csv	FLAG_OWN_CAR
..	...	...
217	installments_payments.csv	NUM_INSTALLMENT_NUMBER
218	installments_payments.csv	DAYS_INSTALLMENT
219	installments_payments.csv	DAYS_ENTRY_PAYMENT
220	installments_payments.csv	AMT_INSTALLMENT
221	installments_payments.csv	AMT_PAYMENT

  

	Description \
1	ID of loan in our sample
2	Target variable (1 - client with payment diffi...
5	Identification if loan is cash or revolving
6	Gender of the client
7	Flag if the client owns a car
..	...
217	On which installment we observe payment
218	When the installment of previous credit was su...
219	When was the installments of previous credit p...
220	What was the prescribed installment amount of ...
221	What the client actually paid on previous cred...

  

	Special
1	NaN
2	NaN
5	NaN
6	NaN
7	NaN
..	...
217	NaN
218	time only relative to the application
219	time only relative to the application
220	NaN
221	NaN

[219 rows x 4 columns]

## 2.1 Tables application\_{train|test}.csv

Il y a plus de 200 colonnes pour 9 tables au format csv ! Avant d'aller plus loin dans l'exploration je vais me concentrer sur les tables principales : les tables application\_{train|test}.csv.

Je vais d'abord regarder les plus grosses corrélations avec la variable TARGET

```
[ ]: train = pd.read_csv('../02_data/application_train.csv')
test = pd.read_csv('../02_data/application_test.csv')
print('Dimensions jeu d\'entraînement :', train.shape)
print('Dimensions jeu de test : ', test.shape)
train.head()
```

Dimensions jeu d'entraînement : (307511, 122)

Dimensions jeu de test : (48744, 121)

```
[ ]: SK_ID_CURR  TARGET NAME_CONTRACT_TYPE CODE_GENDER FLAG_OWN_CAR \
0      100002      1      Cash loans          M          N
1      100003      0      Cash loans          F          N
2      100004      0  Revolving loans          M          Y
3      100006      0      Cash loans          F          N
4      100007      0      Cash loans          M          N

  FLAG_OWN_REALTY  CNT_CHILDREN  AMT_INCOME_TOTAL  AMT_CREDIT  AMT_ANNUITY \
0                Y            0        202500.0    406597.5    24700.5
1                N            0        270000.0   1293502.5    35698.5
2                Y            0         67500.0    135000.0     6750.0
3                Y            0        135000.0    312682.5    29686.5
4                Y            0        121500.0    513000.0    21865.5

... FLAG_DOCUMENT_18 FLAG_DOCUMENT_19 FLAG_DOCUMENT_20 FLAG_DOCUMENT_21 \
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

  AMT_REQ_CREDIT_BUREAU_HOUR  AMT_REQ_CREDIT_BUREAU_DAY \
0                0.0                0.0
1                0.0                0.0
2                0.0                0.0
3                NaN                NaN
4                0.0                0.0

  AMT_REQ_CREDIT_BUREAU_WEEK  AMT_REQ_CREDIT_BUREAU_MON \
0                0.0                0.0
1                0.0                0.0
2                0.0                0.0
3                NaN                NaN
```

4	0.0	0.0
	AMT_REQ_CREDIT_BUREAU_QRT	AMT_REQ_CREDIT_BUREAU_YEAR
0	0.0	1.0
1	0.0	0.0
2	0.0	0.0
3	NaN	NaN
4	0.0	0.0

[5 rows x 122 columns]

```
[ ]: assert len(train.SK_ID_CURR.unique()) == train.shape[0]
      assert len(test.SK_ID_CURR.unique()) == test.shape[0]

      train.set_index('SK_ID_CURR', inplace=True)
      test.set_index('SK_ID_CURR', inplace=True)

      test.head()
```

```
[ ]: NAME_CONTRACT_TYPE CODE_GENDER FLAG_OWN_CAR FLAG_OWN_REALTY \
SK_ID_CURR
100001      Cash loans      F      N      Y
100005      Cash loans      M      N      Y
100013      Cash loans      M      Y      Y
100028      Cash loans      F      N      Y
100038      Cash loans      M      Y      N
```

```
      CNT_CHILDREN  AMT_INCOME_TOTAL  AMT_CREDIT  AMT_ANNUITY \
SK_ID_CURR
100001      0      135000.0      568800.0      20560.5
100005      0      99000.0      222768.0      17370.0
100013      0      202500.0      663264.0      69777.0
100028      2      315000.0      1575000.0      49018.5
100038      1      180000.0      625500.0      32067.0
```

```
      AMT_GOODS_PRICE NAME_TYPE_SUITE ... FLAG_DOCUMENT_18 \
SK_ID_CURR
100001      450000.0  Unaccompanied ...      0
100005      180000.0  Unaccompanied ...      0
100013      630000.0      NaN ...      0
100028      1575000.0  Unaccompanied ...      0
100038      625500.0  Unaccompanied ...      0
```

```
      FLAG_DOCUMENT_19 FLAG_DOCUMENT_20 FLAG_DOCUMENT_21 \
SK_ID_CURR
100001      0      0      0
100005      0      0      0
```

100013	0	0	0
100028	0	0	0
100038	0	0	0

	AMT_REQ_CREDIT_BUREAU_HOUR	AMT_REQ_CREDIT_BUREAU_DAY	\
SK_ID_CURR			
100001	0.0	0.0	
100005	0.0	0.0	
100013	0.0	0.0	
100028	0.0	0.0	
100038	NaN	NaN	

	AMT_REQ_CREDIT_BUREAU_WEEK	AMT_REQ_CREDIT_BUREAU_MON	\
SK_ID_CURR			
100001	0.0	0.0	
100005	0.0	0.0	
100013	0.0	0.0	
100028	0.0	0.0	
100038	NaN	NaN	

	AMT_REQ_CREDIT_BUREAU_QRT	AMT_REQ_CREDIT_BUREAU_YEAR
SK_ID_CURR		
100001	0.0	0.0
100005	0.0	3.0
100013	1.0	4.0
100028	0.0	3.0
100038	NaN	NaN

[5 rows x 120 columns]

```
[ ]: print('name_col' + '\t' + 'data_type' + '\t' + 'dimensionality' + '\t'
        + 'null_count' + '\t' + 'null_perct' + '\t' + 'description')
for col in train.columns.tolist():
    column_tpy = train[col].dtypes
    null_count = train[col].isna().sum()
    null_perct = null_count / train[col].isna().count()
    if train[col].dtype in ['object', 'int64']:
        dimensionality = train[col].nunique()
    else:
        dimensionality = np.nan
    desc = col_desc.loc[col_desc.Table.eq('application_{train|test}.csv')
                      & col_desc.Row.eq(col)].Description.tolist()[0]
    print(col + '\t'
          + str(column_tpy) + '\t'
          + str(dimensionality) + '\t'
          + str(null_count) + '\t'
          + str(round(null_perct, 4) * 100) + '\t')
```

```
+ str(desc))
```

name_col	data_type	dimensionality	null_count	null_perct	
TARGET	int64	2	0	0.0	Target variable (1 - client with payment difficulties: he/she had late payment more than X days on at least one of the first Y installments of the loan in our sample, 0 - all other cases)
NAME_CONTRACT_TYPE	object	2	0	0.0	Identification if loan is cash or revolving
CODE_GENDER	object	3	0	0.0	Gender of the client
FLAG_OWN_CAR	object	2	0	0.0	Flag if the client owns a car
FLAG_OWN_REALTY	object	2	0	0.0	Flag if client owns a house or flat
CNT_CHILDREN	int64	15	0	0.0	Number of children the client has
AMT_INCOME_TOTAL	float64	nan	0	0.0	Income of the client
AMT_CREDIT	float64	nan	0	0.0	Credit amount of the loan
AMT_ANNUITY	float64	nan	12	0.0	Loan annuity
AMT_GOODS_PRICE	float64	nan	278	0.09	For consumer loans it is the price of the goods for which the loan is given
NAME_TYPE_SUITE	object	7	1292	0.42	Who was accompanying client when he was applying for the loan
NAME_INCOME_TYPE	object	8	0	0.0	Clients income type (businessman, working, maternity leave, )
NAME_EDUCATION_TYPE	object	5	0	0.0	Level of highest education the client achieved
NAME_FAMILY_STATUS	object	6	0	0.0	Family status of the client
NAME_HOUSING_TYPE	object	6	0	0.0	What is the housing situation of the client (renting, living with parents, ...)
REGION_POPULATION_RELATIVE	float64	nan	0	0.0	Normalized population of region where client lives (higher number means the client lives in more populated region)
DAYS_BIRTH	int64	17460	0	0.0	Client's age in days at the time of application
DAYS_EMPLOYED	int64	12574	0	0.0	How many days before the application the person started current employment
DAYS_REGISTRATION	float64	nan	0	0.0	How many days before the application did client change his registration
DAYS_ID_PUBLISH	int64	6168	0	0.0	How many days before the application did client change the identity document with which he applied for the loan
OWN_CAR_AGE	float64	nan	202929	65.990000000000001	Age of client's car
FLAG_MOBIL	int64	2	0	0.0	Did client provide mobile phone (1=YES, 0=NO)
FLAG_EMP_PHONE	int64	2	0	0.0	Did client provide work phone

(1=YES, 0=NO)

FLAG_WORK_PHONE	int64	2	0	0.0	Did client provide home phone (1=YES, 0=NO)
FLAG_CONT_MOBILE	int64	2	0	0.0	Was mobile phone reachable (1=YES, 0=NO)
FLAG_PHONE	int64	2	0	0.0	Did client provide home phone (1=YES, 0=NO)
FLAG_EMAIL	int64	2	0	0.0	Did client provide email (1=YES, 0=NO)
OCCUPATION_TYPE	object	18	96391	31.35	What kind of occupation does the client have
CNT_FAM_MEMBERS	float64	nan	2	0.0	How many family members does client have
REGION_RATING_CLIENT	int64	3	0	0.0	Our rating of the region where client lives (1,2,3)
REGION_RATING_CLIENT_W_CITY	int64	3	0	0.0	Our rating of the region where client lives with taking city into account (1,2,3)
WEEKDAY_APPR_PROCESS_START	object	7	0	0.0	On which day of the week did the client apply for the loan
HOURLY_APPR_PROCESS_START	int64	24	0	0.0	Approximately at what hour did the client apply for the loan
REG_REGION_NOT_LIVE_REGION	int64	2	0	0.0	Flag if client's permanent address does not match contact address (1=different, 0=same, at region level)
REG_REGION_NOT_WORK_REGION	int64	2	0	0.0	Flag if client's permanent address does not match work address (1=different, 0=same, at region level)
LIVE_REGION_NOT_WORK_REGION	int64	2	0	0.0	Flag if client's contact address does not match work address (1=different, 0=same, at region level)
REG_CITY_NOT_LIVE_CITY	int64	2	0	0.0	Flag if client's permanent address does not match contact address (1=different, 0=same, at city level)
REG_CITY_NOT_WORK_CITY	int64	2	0	0.0	Flag if client's permanent address does not match work address (1=different, 0=same, at city level)
LIVE_CITY_NOT_WORK_CITY	int64	2	0	0.0	Flag if client's contact address does not match work address (1=different, 0=same, at city level)
ORGANIZATION_TYPE	object	58	0	0.0	Type of organization where client works
EXT_SOURCE_1	float64	nan	173378	56.379999999999995	Normalized score from external data source
EXT_SOURCE_2	float64	nan	660	0.21	Normalized score from external data source
EXT_SOURCE_3	float64	nan	60965	19.830000000000002	Normalized score from external data source
APARTMENTS_AVG	float64	nan	156061	50.74999999999999	Normalized information about building where the client lives, What is average (_AVG

suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

BASEMENTAREA\_AVG float64 nan 179943 58.52 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

YEARS\_BEGINEXPLUATATION\_AVG float64 nan 150007 48.78 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

YEARS\_BUILD\_AVG float64 nan 204488 66.5 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

COMMONAREA\_AVG float64 nan 214865 69.87 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

ELEVATORS\_AVG float64 nan 163891 53.300000000000004 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

ENTRANCES\_AVG float64 nan 154828 50.349999999999994 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

FLOORSMAX\_AVG float64 nan 153020 49.76 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

FLOORSMIN\_AVG float64 nan 208642 67.85 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

LANDAREA\_AVG float64 nan 182590 59.38 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor



LIVINGAPARTMENTS\_AVG float64 nan 210199 68.35 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

LIVINGAREA\_AVG float64 nan 154350 50.19 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

NONLIVINGAPARTMENTS\_AVG float64 nan 213514 69.43 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

NONLIVINGAREA\_AVG float64 nan 169682 55.179999999999999 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

APARTMENTS\_MODE float64 nan 156061 50.749999999999999 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

BASEMENTAREA\_MODE float64 nan 179943 58.52 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

YEARS\_BEGINEXPLUATATION\_MODE float64 nan 150007 48.78 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

YEARS\_BUILD\_MODE float64 nan 204488 66.5 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

COMMONAREA\_MODE float64 nan 214865 69.87 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

ELEVATORS\_MODE float64 nan 163891 53.300000000000004 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common

area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

ENTRANCES\_MODE float64 nan 154828 50.349999999999994 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

FLOORSMAX\_MODE float64 nan 153020 49.76 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

FLOORSMIN\_MODE float64 nan 208642 67.85 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

LANDAREA\_MODE float64 nan 182590 59.38 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

LIVINGAPARTMENTS\_MODE float64 nan 210199 68.35 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

LIVINGAREA\_MODE float64 nan 154350 50.19 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

NONLIVINGAPARTMENTS\_MODE float64 nan 213514 69.43 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

NONLIVINGAREA\_MODE float64 nan 169682 55.179999999999999 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

APARTMENTS\_MEDI float64 nan 156061 50.749999999999999 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

BASEMENTAREA\_MEDI float64 nan 179943 58.52 Normalized information

about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

YEARS\_BEGINEXPLUATATION\_MEDI float64 nan 150007 48.78 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

YEARS\_BUILD\_MEDI float64 nan 204488 66.5 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

COMMONAREA\_MEDI float64 nan 214865 69.87 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

ELEVATORS\_MEDI float64 nan 163891 53.300000000000004 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

ENTRANCES\_MEDI float64 nan 154828 50.349999999999994 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

FLOORSMAX\_MEDI float64 nan 153020 49.76 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

FLOORSMIN\_MEDI float64 nan 208642 67.85 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

LANDAREA\_MEDI float64 nan 182590 59.38 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

LIVINGAPARTMENTS\_MEDI float64 nan 210199 68.35 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the

building, number of floor

LIVINGAREA\_MEDI float64 nan 154350 50.19 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

NONLIVINGAPARTMENTS\_MEDI float64 nan 213514 69.43 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

NONLIVINGAREA\_MEDI float64 nan 169682 55.17999999999999 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

FONDKAPREMONT\_MODE object 4 210295 68.39 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

HOUSETYPE\_MODE object 3 154297 50.18 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

TOTALAREA\_MODE float64 nan 148431 48.27 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

WALLSMATERIAL\_MODE object 7 156341 50.839999999999996 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

EMERGENCYSTATE\_MODE object 2 145755 47.4 Normalized information about building where the client lives, What is average (\_AVG suffix), modus (\_MODE suffix), median (\_MEDI suffix) apartment size, common area, living area, age of building, number of elevators, number of entrances, state of the building, number of floor

OBS\_30\_CNT\_SOCIAL\_CIRCLE float64 nan 1021 0.33 How many observation of client's social surroundings with observable 30 DPD (days past due) default

DEF\_30\_CNT\_SOCIAL\_CIRCLE float64 nan 1021 0.33 How many observation of client's social surroundings defaulted on 30 DPD (days past due)

OBS\_60\_CNT\_SOCIAL\_CIRCLE float64 nan 1021 0.33 How many observation of client's social surroundings with observable 60 DPD (days past

```

due) default
DEF_60_CNT_SOCIAL_CIRCLE      float64 nan    1021    0.33    How many
observation of client's social surroundings defaulted on 60 (days past due) DPD
DAYS_LAST_PHONE_CHANGE float64 nan    1    0.0    How many days before
application did client change phone
FLAG_DOCUMENT_2 int64    2    0    0.0    Did client provide document 2
FLAG_DOCUMENT_3 int64    2    0    0.0    Did client provide document 3
FLAG_DOCUMENT_4 int64    2    0    0.0    Did client provide document 4
FLAG_DOCUMENT_5 int64    2    0    0.0    Did client provide document 5
FLAG_DOCUMENT_6 int64    2    0    0.0    Did client provide document 6
FLAG_DOCUMENT_7 int64    2    0    0.0    Did client provide document 7
FLAG_DOCUMENT_8 int64    2    0    0.0    Did client provide document 8
FLAG_DOCUMENT_9 int64    2    0    0.0    Did client provide document 9
FLAG_DOCUMENT_10 int64    2    0    0.0    Did client provide
document 10
FLAG_DOCUMENT_11 int64    2    0    0.0    Did client provide
document 11
FLAG_DOCUMENT_12 int64    2    0    0.0    Did client provide
document 12
FLAG_DOCUMENT_13 int64    2    0    0.0    Did client provide
document 13
FLAG_DOCUMENT_14 int64    2    0    0.0    Did client provide
document 14
FLAG_DOCUMENT_15 int64    2    0    0.0    Did client provide
document 15
FLAG_DOCUMENT_16 int64    2    0    0.0    Did client provide
document 16
FLAG_DOCUMENT_17 int64    2    0    0.0    Did client provide
document 17
FLAG_DOCUMENT_18 int64    2    0    0.0    Did client provide
document 18
FLAG_DOCUMENT_19 int64    2    0    0.0    Did client provide
document 19
FLAG_DOCUMENT_20 int64    2    0    0.0    Did client provide
document 20
FLAG_DOCUMENT_21 int64    2    0    0.0    Did client provide
document 21
AMT_REQ_CREDIT_BUREAU_HOUR float64 nan    41519    13.5    Number of
enquiries to Credit Bureau about the client one hour before application
AMT_REQ_CREDIT_BUREAU_DAY float64 nan    41519    13.5    Number of
enquiries to Credit Bureau about the client one day before application
(excluding one hour before application)
AMT_REQ_CREDIT_BUREAU_WEEK float64 nan    41519    13.5    Number of
enquiries to Credit Bureau about the client one week before application
(excluding one day before application)
AMT_REQ_CREDIT_BUREAU_MON float64 nan    41519    13.5    Number of
enquiries to Credit Bureau about the client one month before application
(excluding one week before application)

```

AMT_REQ_CREDIT_BUREAU_QRT	float64	nan	41519	13.5	Number of enquiries to Credit Bureau about the client 3 month before application (excluding one month before application)
AMT_REQ_CREDIT_BUREAU_YEAR	float64	nan	41519	13.5	Number of enquiries to Credit Bureau about the client one day year (excluding last 3 months before application)

### 2.1.1 Encodage des colonnes textuelles

```
[ ]: print(train.dtypes.value_counts())
```

```
float64    65
int64      40
object     16
dtype: int64
```

```
[ ]: categor_feats = train.select_dtypes('object').columns.tolist()
print(categor_feats)
```

```
['NAME_CONTRACT_TYPE', 'CODE_GENDER', 'FLAG_OWN_CAR', 'FLAG_OWN_REALTY',
'NAME_TYPE_SUITE', 'NAME_INCOME_TYPE', 'NAME_EDUCATION_TYPE',
'NAME_FAMILY_STATUS', 'NAME_HOUSING_TYPE', 'OCCUPATION_TYPE',
'WEEKDAY_APPR_PROCESS_START', 'ORGANIZATION_TYPE', 'FONDKAPREMONT_MODE',
'HOUSETYPE_MODE', 'WALLSMATERIAL_MODE', 'EMERGENCYSTATE_MODE']
```

```
[ ]: print(train[categor_feats].apply(pd.Series.nunique,
                                     axis=0).sort_values(ascending=False))
```

```
ORGANIZATION_TYPE    58
OCCUPATION_TYPE      18
NAME_INCOME_TYPE      8
NAME_TYPE_SUITE       7
WEEKDAY_APPR_PROCESS_START  7
WALLSMATERIAL_MODE    7
NAME_FAMILY_STATUS    6
NAME_HOUSING_TYPE      6
NAME_EDUCATION_TYPE    5
FONDKAPREMONT_MODE     4
CODE_GENDER           3
HOUSETYPE_MODE         3
NAME_CONTRACT_TYPE     2
FLAG_OWN_CAR           2
FLAG_OWN_REALTY         2
EMERGENCYSTATE_MODE    2
dtype: int64
```

### Encodage des catégories multi-dimensionnelles

```
[ ]: dimensionality = lambda x,df : df[[x]].apply(pd.Series.nunique).values
```

```
categor_feats_multidim = []
for feat in categor_feats:
    if dimensionality(feat,train) > 2:
        categor_feats_multidim.append(feat)
print(categor_feats_multidim)
```

```
['CODE_GENDER', 'NAME_TYPE_SUITE', 'NAME_INCOME_TYPE', 'NAME_EDUCATION_TYPE',
'NAME_FAMILY_STATUS', 'NAME_HOUSING_TYPE', 'OCCUPATION_TYPE',
'WEEKDAY_APPR_PROCESS_START', 'ORGANIZATION_TYPE', 'FONDKAPREMONT_MODE',
'HOUSETYPE_MODE', 'WALLSMATERIAL_MODE']
```

```
[ ]: for feat in categor_feats_multidim:
    if dimensionality(feat,train) <= 8:
        print(feat, train[feat].unique())
```

```
CODE_GENDER ['M' 'F' 'XNA']
NAME_TYPE_SUITE ['Unaccompanied' 'Family' 'Spouse, partner' 'Children' 'Other_A'
nan
'Other_B' 'Group of people']
NAME_INCOME_TYPE ['Working' 'State servant' 'Commercial associate' 'Pensioner'
'Unemployed'
'Student' 'Businessman' 'Maternity leave']
NAME_EDUCATION_TYPE ['Secondary / secondary special' 'Higher education'
'Incomplete higher'
'Lower secondary' 'Academic degree']
NAME_FAMILY_STATUS ['Single / not married' 'Married' 'Civil marriage' 'Widow'
'Separated'
'Unknown']
NAME_HOUSING_TYPE ['House / apartment' 'Rented apartment' 'With parents'
'Municipal apartment' 'Office apartment' 'Co-op apartment']
WEEKDAY_APPR_PROCESS_START ['WEDNESDAY' 'MONDAY' 'THURSDAY' 'SUNDAY' 'SATURDAY'
'FRIDAY' 'TUESDAY']
FONDKAPREMONT_MODE ['reg oper account' nan 'org spec account' 'reg oper spec
account'
'not specified']
HOUSETYPE_MODE ['block of flats' nan 'terraced house' 'specific housing']
WALLSMATERIAL_MODE ['Stone, brick' 'Block' nan 'Panel' 'Mixed' 'Wooden' 'Others'
'Monolithic']
```

On remarque deux choses : 1. La variable CODE\_GENDER n'est pas vraiment multidimensionnelle 2. La variable WEEKDAY\_APPR\_START est catégorique ordinaire, et devrait être traitée à part en tant que variable de temps 3. Dans les autres variables, il y a des espaces et des caractères spéciaux qu'il va falloir remplacer si on veut récupérer des dummy variables avec des noms simples à manipuler

On peut procéder comme ça : \* pour les espaces () : remplacer par des \_ \* pour les / ou les , : remplacer par des or \* pour toutes les variables, inclure le nom de la variable au début de chaque valeur

```
[ ]: categor_feats_multidim.remove('CODE_GENDER')
categor_feats_multidim.remove('WEEKDAY_APPR_PROCESS_START')
train_categor_multidim = train[categor_feats_multidim]
train_categor_multidim
```

```
[ ]:      NAME_TYPE_SUITE      NAME_INCOME_TYPE  \
SK_ID_CURR
100002      Unaccompanied      Working
100003      Family      State servant
100004      Unaccompanied      Working
100006      Unaccompanied      Working
100007      Unaccompanied      Working
...      ...      ...
456251      Unaccompanied      Working
456252      Unaccompanied      Pensioner
456253      Unaccompanied      Working
456254      Unaccompanied      Commercial associate
456255      Unaccompanied      Commercial associate

      NAME_EDUCATION_TYPE      NAME_FAMILY_STATUS  \
SK_ID_CURR
100002      Secondary / secondary special      Single / not married
100003      Higher education      Married
100004      Secondary / secondary special      Single / not married
100006      Secondary / secondary special      Civil marriage
100007      Secondary / secondary special      Single / not married
...      ...      ...
456251      Secondary / secondary special      Separated
456252      Secondary / secondary special      Widow
456253      Higher education      Separated
456254      Secondary / secondary special      Married
456255      Higher education      Married

      NAME_HOUSING_TYPE      OCCUPATION_TYPE      ORGANIZATION_TYPE  \
SK_ID_CURR
100002      House / apartment      Laborers      Business Entity Type 3
100003      House / apartment      Core staff      School
100004      House / apartment      Laborers      Government
100006      House / apartment      Laborers      Business Entity Type 3
100007      House / apartment      Core staff      Religion
...      ...      ...      ...
456251      With parents      Sales staff      Services
456252      House / apartment      NaN      XNA
456253      House / apartment      Managers      School
456254      House / apartment      Laborers      Business Entity Type 1
456255      House / apartment      Laborers      Business Entity Type 3
```



SK_ID_CURR	FONDKAPREMONT_MODE	HOUSETYPE_MODE	WALLSMATERIAL_MODE
100002	reg oper account	block of flats	Stone, brick
100003	reg oper account	block of flats	Block
100004	NaN	NaN	NaN
100006	NaN	NaN	NaN
100007	NaN	NaN	NaN
...	...	...	...
456251	reg oper account	block of flats	Stone, brick
456252	reg oper account	block of flats	Stone, brick
456253	reg oper account	block of flats	Panel
456254	NaN	block of flats	Stone, brick
456255	NaN	block of flats	Panel

[307511 rows x 10 columns]

```
[ ]: def format_categor_values(x):
    y = x.lower()
    y = y.replace(' ', '_')
    y = y.replace('-', '').replace(':', '')
    y = y.replace(',', '_or').replace('/', 'or')
    return y

print(train.NAME_HOUSING_TYPE.apply(format_categor_values).value_counts())

format_vfunc = np.vectorize(format_categor_values)
categor_value_formatter = FunctionTransformer(lambda x: format_vfunc(x))

#concat_feat_name_with_value = lambda x: '___' + x.name + '_' + x.astype(str)
print(categor_value_formatter.fit_transform(train.NAME_HOUSING_TYPE))
```

```
house_or_apartment      272868
with_parents            14840
municipal_apartment     11183
rented_apartment        4881
office_apartment        2617
coop_apartment          1122
Name: NAME_HOUSING_TYPE, dtype: int64
['house_or_apartment' 'house_or_apartment' 'house_or_apartment' ...
 'house_or_apartment' 'house_or_apartment' 'house_or_apartment']
```

```
[ ]: categor_multidim_preprocessor = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='constant', fill_value='Unknown')),
    ('value_formatter', categor_value_formatter),
    ('encoder', OneHotEncoder())])
categor_multidim_preprocessor.fit_transform(train_categor_multidim)
```

```
[ ]: <307511x127 sparse matrix of type '<class 'numpy.float64'>'
      with 3075110 stored elements in Compressed Sparse Row format>
```

```
[ ]: feat_name_replacement = {k:v for k,v in zip(range(len(categor_feats_multidim)),
                                                categor_feats_multidim)}

onehot_feat_names = []
for feat_name in [n.replace('encoder_x', '')\
                  for n in get_feature_names(categor_multidim_preprocessor)]:
    for i in range(len(categor_feats_multidim)):
        if feat_name[0] == str(i):
            new_feat_name = feat_name_replacement[i] + feat_name[1:]
            onehot_feat_names.append(new_feat_name)

print(onehot_feat_names)
```

```
['NAME_TYPE_SUITE_children', 'NAME_TYPE_SUITE_family',
'NAME_TYPE_SUITE_group_of_people', 'NAME_TYPE_SUITE_other_a',
'NAME_TYPE_SUITE_other_b', 'NAME_TYPE_SUITE_spouse_or_partner',
'NAME_TYPE_SUITE_unaccompanied', 'NAME_TYPE_SUITE_unknown',
'NAME_INCOME_TYPE_businessman', 'NAME_INCOME_TYPE_commercial_associate',
'NAME_INCOME_TYPE_maternity_leave', 'NAME_INCOME_TYPE_pensioner',
'NAME_INCOME_TYPE_state_servant', 'NAME_INCOME_TYPE_student',
'NAME_INCOME_TYPE_unemployed', 'NAME_INCOME_TYPE_working',
'NAME_EDUCATION_TYPE_academic_degree', 'NAME_EDUCATION_TYPE_higher_education',
'NAME_EDUCATION_TYPE_incomplete_higher', 'NAME_EDUCATION_TYPE_lower_secondary',
'NAME_EDUCATION_TYPE_secondary_or_secondary_special',
'NAME_FAMILY_STATUS_civil_marriage', 'NAME_FAMILY_STATUS_married',
'NAME_FAMILY_STATUS_separated', 'NAME_FAMILY_STATUS_single_or_not_married',
'NAME_FAMILY_STATUS_unknown', 'NAME_FAMILY_STATUS_widow',
'NAME_HOUSING_TYPE_coop_apartment', 'NAME_HOUSING_TYPE_house_or_apartment',
'NAME_HOUSING_TYPE_municipal_apartment', 'NAME_HOUSING_TYPE_office_apartment',
'NAME_HOUSING_TYPE_rented_apartment', 'NAME_HOUSING_TYPE_with_parents',
'OCCUPATION_TYPE_accountants', 'OCCUPATION_TYPE_cleaning_staff',
'OCCUPATION_TYPE_cooking_staff', 'OCCUPATION_TYPE_core_staff',
'OCCUPATION_TYPE_drivers', 'OCCUPATION_TYPE_high_skill_tech_staff',
'OCCUPATION_TYPE_hr_staff', 'OCCUPATION_TYPE_it_staff',
'OCCUPATION_TYPE_laborers', 'OCCUPATION_TYPE_lowskill_laborers',
'OCCUPATION_TYPE_managers', 'OCCUPATION_TYPE_medicine_staff',
'OCCUPATION_TYPE_private_service_staff', 'OCCUPATION_TYPE_realty_agents',
'OCCUPATION_TYPE_sales_staff', 'OCCUPATION_TYPE_secretaries',
'OCCUPATION_TYPE_security_staff', 'OCCUPATION_TYPE_unknown',
'OCCUPATION_TYPE_waitersorbarmen_staff', 'ORGANIZATION_TYPE_advertising',
'ORGANIZATION_TYPE_agriculture', 'ORGANIZATION_TYPE_bank',
'ORGANIZATION_TYPE_business_entity_type_1',
'ORGANIZATION_TYPE_business_entity_type_2',
'ORGANIZATION_TYPE_business_entity_type_3', 'ORGANIZATION_TYPE_cleaning',
'ORGANIZATION_TYPE_construction', 'ORGANIZATION_TYPE_culture',
```

```

'ORGANIZATION_TYPE_electricity', 'ORGANIZATION_TYPE_emergency',
'ORGANIZATION_TYPE_government', 'ORGANIZATION_TYPE_hotel',
'ORGANIZATION_TYPE_housing', 'ORGANIZATION_TYPE_industry_type_1',
'ORGANIZATION_TYPE_industry_type_10', 'ORGANIZATION_TYPE_industry_type_11',
'ORGANIZATION_TYPE_industry_type_12', 'ORGANIZATION_TYPE_industry_type_13',
'ORGANIZATION_TYPE_industry_type_2', 'ORGANIZATION_TYPE_industry_type_3',
'ORGANIZATION_TYPE_industry_type_4', 'ORGANIZATION_TYPE_industry_type_5',
'ORGANIZATION_TYPE_industry_type_6', 'ORGANIZATION_TYPE_industry_type_7',
'ORGANIZATION_TYPE_industry_type_8', 'ORGANIZATION_TYPE_industry_type_9',
'ORGANIZATION_TYPE_insurance', 'ORGANIZATION_TYPE_kindergarten',
'ORGANIZATION_TYPE_legal_services', 'ORGANIZATION_TYPE_medicine',
'ORGANIZATION_TYPE_military', 'ORGANIZATION_TYPE_mobile',
'ORGANIZATION_TYPE_other', 'ORGANIZATION_TYPE_police',
'ORGANIZATION_TYPE_postal', 'ORGANIZATION_TYPE_realtor',
'ORGANIZATION_TYPE_religion', 'ORGANIZATION_TYPE_restaurant',
'ORGANIZATION_TYPE_school', 'ORGANIZATION_TYPE_security',
'ORGANIZATION_TYPE_security_ministries', 'ORGANIZATION_TYPE_selfemployed',
'ORGANIZATION_TYPE_services', 'ORGANIZATION_TYPE_telecom',
'ORGANIZATION_TYPE_trade_type_1', 'ORGANIZATION_TYPE_trade_type_2',
'ORGANIZATION_TYPE_trade_type_3', 'ORGANIZATION_TYPE_trade_type_4',
'ORGANIZATION_TYPE_trade_type_5', 'ORGANIZATION_TYPE_trade_type_6',
'ORGANIZATION_TYPE_trade_type_7', 'ORGANIZATION_TYPE_transport_type_1',
'ORGANIZATION_TYPE_transport_type_2', 'ORGANIZATION_TYPE_transport_type_3',
'ORGANIZATION_TYPE_transport_type_4', 'ORGANIZATION_TYPE_university',
'ORGANIZATION_TYPE_xna', 'FONDKAPREMONT_MODE_not_specified',
'FONDKAPREMONT_MODE_org_spec_account', 'FONDKAPREMONT_MODE_reg_oper_account',
'FONDKAPREMONT_MODE_reg_oper_spec_account', 'FONDKAPREMONT_MODE_unknown',
'HOUSETYPE_MODE_block_of_flats', 'HOUSETYPE_MODE_specific_housing',
'HOUSETYPE_MODE_terraced_house', 'HOUSETYPE_MODE_unknown',
'WALLSMATERIAL_MODE_block', 'WALLSMATERIAL_MODE_mixed',
'WALLSMATERIAL_MODE_monolithic', 'WALLSMATERIAL_MODE_others',
'WALLSMATERIAL_MODE_panel', 'WALLSMATERIAL_MODE_stone_or_brick',
'WALLSMATERIAL_MODE_unknown', 'WALLSMATERIAL_MODE_wooden']

```

### Encodage des catégories bi-dimensionnelles

```

[ ]: categor_feats_binary = []
for feat in categor_feats:
    if dimensionality(feat,train) <= 2:
        categor_feats_binary.append(feat)
print(categor_feats_binary)

```

```
['NAME_CONTRACT_TYPE', 'FLAG_OWN_CAR', 'FLAG_OWN_REALTY', 'EMERGENCYSTATE_MODE']
```

```

[ ]: categor_feats_binary.append('CODE_GENDER')
categor_feats_binary.append('WEEKDAY_APPR_PROCESS_START')
for feat in categor_feats_binary:
    print(feat, train[feat].unique())

```

```
NAME_CONTRACT_TYPE ['Cash loans' 'Revolving loans']
FLAG_OWN_CAR ['N' 'Y']
FLAG_OWN_REALTY ['Y' 'N']
EMERGENCYSTATE_MODE ['No' nan 'Yes']
CODE_GENDER ['M' 'F' 'XNA']
WEEKDAY_APPR_PROCESS_START ['WEDNESDAY' 'MONDAY' 'THURSDAY' 'SUNDAY' 'SATURDAY'
'FRIDAY' 'TUESDAY']
```

```
[ ]: train_categor_binary = train[categor_feats_binary]

contract_types = ['Cash loans', 'Revolving loans']
y_or_n = ['N', 'Y']
yes_or_no = ['No', 'Yes']
genders = ['M', 'F']
weekdays = ['MONDAY', 'TUESDAY', 'WEDNESDAY', 'THURSDAY', 'FRIDAY', 'SATURDAY',
             'SUNDAY']
categories = [contract_types, y_or_n, y_or_n, yes_or_no, genders, weekdays]

categor_binary_preprocessor = Pipeline(steps=[
    ('nan_imputer', SimpleImputer(strategy='most_frequent')),
    ('xna_imputer', SimpleImputer(missing_values='XNA',
                                   strategy='most_frequent')),
    ('encoder', OrdinalEncoder(categories=categories))]
)
categor_binary_preprocessor.fit_transform(train_categor_binary)
```

```
[ ]: array([[0., 0., 1., 0., 0., 2.],
           [0., 0., 0., 0., 1., 0.],
           [1., 1., 1., 0., 0., 0.],
           ...,
           [0., 0., 1., 0., 1., 3.],
           [0., 0., 1., 0., 1., 2.],
           [0., 0., 0., 0., 1., 3.]])
```

### Pipeline finale des variables catégoriques

```
[ ]: categor_preprocessor = make_column_transformer(
    (categor_binary_preprocessor, categor_feats_binary),
    (categor_multidim_preprocessor, categor_feats_multidim),
    remainder='passthrough'
)

categor_preprocessor.fit_transform(train)
```

```
[ ]: array([[0., 0., 1., ..., 0., 0., 1.],
           [0., 0., 0., ..., 0., 0., 0.],
           [1., 1., 1., ..., 0., 0., 0.],
           ...,
           [0., 0., 1., ..., 1., 0., 1.]])
```

```
[0., 0., 1., ..., 0., 0., 0.],
[0., 0., 0., ..., 2., 0., 1.]])
```

```
[ ]: train_encoded = categor_preprocessor.fit_transform(train)
print(train_encoded.shape)
```

```
(307511, 238)
```

```
[ ]: binary_feat_names = get_feature_names(categor_binary_preprocessor)
print(binary_feat_names)
```

```
[]
```

```
[ ]: train_encoded[:5]
```

```
[ ]: array([[ 0.,  0.,  1., ...,  0.,  0.,  1.],
[ 0.,  0.,  0., ...,  0.,  0.,  0.],
[ 1.,  1.,  1., ...,  0.,  0.,  0.],
[ 0.,  0.,  1., ..., nan, nan, nan],
[ 0.,  0.,  1., ...,  0.,  0.,  0.]])
```

### 2.1.2 Pré-traitement des variables numériques

```
[ ]: numeric_feats = train.select_dtypes(['int64', 'float64']).columns.tolist()
print(numeric_feats)
```

```
['TARGET', 'CNT_CHILDREN', 'AMT_INCOME_TOTAL', 'AMT_CREDIT', 'AMT_ANNUITY',
'AMT_GOODS_PRICE', 'REGION_POPULATION_RELATIVE', 'DAYS_BIRTH', 'DAYS_EMPLOYED',
'DAYS_REGISTRATION', 'DAYS_ID_PUBLISH', 'OWN_CAR_AGE', 'FLAG_MOBIL',
'FLAG_EMP_PHONE', 'FLAG_WORK_PHONE', 'FLAG_CONT_MOBILE', 'FLAG_PHONE',
'FLAG_EMAIL', 'CNT_FAM_MEMBERS', 'REGION_RATING_CLIENT',
'REGION_RATING_CLIENT_W_CITY', 'HOUR_APPR_PROCESS_START',
'REG_REGION_NOT_LIVE_REGION', 'REG_REGION_NOT_WORK_REGION',
'LIVE_REGION_NOT_WORK_REGION', 'REG_CITY_NOT_LIVE_CITY',
'REG_CITY_NOT_WORK_CITY', 'LIVE_CITY_NOT_WORK_CITY', 'EXT_SOURCE_1',
'EXT_SOURCE_2', 'EXT_SOURCE_3', 'APARTMENTS_AVG', 'BASEMENTAREA_AVG',
'YEARS_BEGINEXPLUATATION_AVG', 'YEARS_BUILD_AVG', 'COMMONAREA_AVG',
'ELEVATORS_AVG', 'ENTRANCES_AVG', 'FLOORSMAX_AVG', 'FLOORSMIN_AVG',
'LANDAREA_AVG', 'LIVINGAPARTMENTS_AVG', 'LIVINGAREA_AVG',
'NONLIVINGAPARTMENTS_AVG', 'NONLIVINGAREA_AVG', 'APARTMENTS_MODE',
'BASEMENTAREA_MODE', 'YEARS_BEGINEXPLUATATION_MODE', 'YEARS_BUILD_MODE',
'COMMONAREA_MODE', 'ELEVATORS_MODE', 'ENTRANCES_MODE', 'FLOORSMAX_MODE',
'FLOORSMIN_MODE', 'LANDAREA_MODE', 'LIVINGAPARTMENTS_MODE', 'LIVINGAREA_MODE',
'NONLIVINGAPARTMENTS_MODE', 'NONLIVINGAREA_MODE', 'APARTMENTS_MEDI',
'BASEMENTAREA_MEDI', 'YEARS_BEGINEXPLUATATION_MEDI', 'YEARS_BUILD_MEDI',
'COMMONAREA_MEDI', 'ELEVATORS_MEDI', 'ENTRANCES_MEDI', 'FLOORSMAX_MEDI',
'FLOORSMIN_MEDI', 'LANDAREA_MEDI', 'LIVINGAPARTMENTS_MEDI', 'LIVINGAREA_MEDI',
```

```
'NONLIVINGAPARTMENTS_MEDI', 'NONLIVINGAREA_MEDI', 'TOTALAREA_MODE',
'OBS_30_CNT_SOCIAL_CIRCLE', 'DEF_30_CNT_SOCIAL_CIRCLE',
'OBS_60_CNT_SOCIAL_CIRCLE', 'DEF_60_CNT_SOCIAL_CIRCLE',
'DAYS_LAST_PHONE_CHANGE', 'FLAG_DOCUMENT_2', 'FLAG_DOCUMENT_3',
'FLAG_DOCUMENT_4', 'FLAG_DOCUMENT_5', 'FLAG_DOCUMENT_6', 'FLAG_DOCUMENT_7',
'FLAG_DOCUMENT_8', 'FLAG_DOCUMENT_9', 'FLAG_DOCUMENT_10', 'FLAG_DOCUMENT_11',
'FLAG_DOCUMENT_12', 'FLAG_DOCUMENT_13', 'FLAG_DOCUMENT_14', 'FLAG_DOCUMENT_15',
'FLAG_DOCUMENT_16', 'FLAG_DOCUMENT_17', 'FLAG_DOCUMENT_18', 'FLAG_DOCUMENT_19',
'FLAG_DOCUMENT_20', 'FLAG_DOCUMENT_21', 'AMT_REQ_CREDIT_BUREAU_HOUR',
'AMT_REQ_CREDIT_BUREAU_DAY', 'AMT_REQ_CREDIT_BUREAU_WEEK',
'AMT_REQ_CREDIT_BUREAU_MON', 'AMT_REQ_CREDIT_BUREAU_QRT',
'AMT_REQ_CREDIT_BUREAU_YEAR']
```

```
[ ]:
```

```
[ ]: desc = col_desc.loc[col_desc.Table.eq('application_{train|test}.csv')
& col_desc.Row.eq(col)].Description.tolist()
```

```
['What kind of occupation does the client have']
```

```
[ ]: print(col_desc.loc[col_desc.Table.eq('application_{train|test}.csv')
& col_desc.Row.eq('CODE_GENDER')].Description.tolist())
```

```
['Gender of the client']
```

### 2.1.3 Étude des variables

```
[ ]: train.EXT_SOURCE_1.dtype
```

```
[ ]: dtype('float64')
```

```
[ ]: train.NAME_FAMILY_STATUS.value_counts()
```

```
[ ]: Married          196432
Single / not married  45444
Civil marriage       29775
Separated            19770
Widow                16088
Unknown              2
Name: NAME_FAMILY_STATUS, dtype: int64
```

```
[ ]: train.NAME_EDUCATION_TYPE.value_counts()
```

```
[ ]: Secondary / secondary special  218391
Higher education                   74863
Incomplete higher                  10277
Lower secondary                    3816
Academic degree                    164
```

Name: NAME\_EDUCATION\_TYPE, dtype: int64

```
[ ]: train.NAME_TYPE_SUITE.value_counts()
```

```
[ ]: Unaccompanied      248526
      Family            40149
      Spouse, partner    11370
      Children          3267
      Other_B           1770
      Other_A           866
      Group of people    271
      Name: NAME_TYPE_SUITE, dtype: int64
```

```
[ ]: train['AGE'] = round(train['DAYS_BIRTH'] / - 365, 0).astype('int')
```

```
[ ]: train.AGE
```

```
[ ]: 0      26
      1      46
      2      52
      3      52
      4      55
      ..
      307506  26
      307507  57
      307508  41
      307509  33
      307510  46
      Name: AGE, Length: 307511, dtype: int64
```

## 2.2 Autres tables

```
[ ]: col_desc.loc[col_desc.Table == 'bureau_balance.csv'].Description.values
```

```
[ ]: array(['Recoded ID of Credit Bureau credit (unique coding for each application)
- use this to join to CREDIT_BUREAU table ',
      'Month of balance relative to application date (-1 means the freshest
balance date)',
      'Status of Credit Bureau loan during the month (active, closed, DPD0-30,
[C means closed, X means status unknown, 0 means no DPD, 1 means maximal did
during month between 1-30, 2 means DPD 31-60, 5 means DPD 120+ or sold or
written off ] )'],
      dtype=object)
```

```
[ ]: bureau = pd.read_csv('../02_data/bureau.csv')
      bureau_balance = pd.read_csv('../02_data/bureau_balance.csv')
```

```
[ ]: bureau_balance.shape
```

```
[ ]: (27299925, 3)
```

```
[ ]: bureau_balance.shape[0] / 10 ** 3
```

```
[ ]: 27299.925
```

```
[ ]: bureau_balance.columns
```

```
[ ]: Index(['SK_ID_BUREAU', 'MONTHS_BALANCE', 'STATUS'], dtype='object')
```

```
[ ]: bureau.shape
```

```
[ ]: (1716428, 17)
```

```
[ ]: bureau = pd.read_csv('../02_data/bureau.csv')  
bureau
```

```
[ ]:      SK_ID_CURR  SK_ID_BUREAU  CREDIT_ACTIVE  CREDIT_CURRENCY  DAYS_CREDIT  \  
0          215354      5714462         Closed      currency 1         -497  
1          215354      5714463         Active      currency 1         -208  
2          215354      5714464         Active      currency 1         -203  
3          215354      5714465         Active      currency 1         -203  
4          215354      5714466         Active      currency 1         -629  
...          ...          ...          ...          ...          ...  
1716423      259355      5057750         Active      currency 1          -44  
1716424      100044      5057754         Closed      currency 1        -2648  
1716425      100044      5057762         Closed      currency 1        -1809  
1716426      246829      5057770         Closed      currency 1        -1878  
1716427      246829      5057778         Closed      currency 1         -463
```

```
      CREDIT_DAY_OVERDUE  DAYS_CREDIT_ENDDATE  DAYS_ENDDATE_FACT  \  
0              0          -153.0          -153.0  
1              0          1075.0             NaN  
2              0           528.0             NaN  
3              0             NaN             NaN  
4              0          1197.0             NaN  
...          ...          ...          ...  
1716423          0          -30.0             NaN  
1716424          0         -2433.0        -2493.0  
1716425          0         -1628.0         -970.0  
1716426          0         -1513.0        -1513.0  
1716427          0             NaN        -387.0
```

```
      AMT_CREDIT_MAX_OVERDUE  CNT_CREDIT_PROLONG  AMT_CREDIT_SUM  \  
0              NaN              0          91323.00
```



1	NaN	0	225000.00
2	NaN	0	464323.50
3	NaN	0	90000.00
4	77674.5	0	2700000.00
...	...	...	...
1716423	0.0	0	11250.00
1716424	5476.5	0	38130.84
1716425	NaN	0	15570.00
1716426	NaN	0	36000.00
1716427	NaN	0	22500.00

	AMT_CREDIT_SUM_DEBT	AMT_CREDIT_SUM_LIMIT	AMT_CREDIT_SUM_OVERDUE \
0	0.0	NaN	0.0
1	171342.0	NaN	0.0
2	NaN	NaN	0.0
3	NaN	NaN	0.0
4	NaN	NaN	0.0
...	...	...	...
1716423	11250.0	0.0	0.0
1716424	0.0	0.0	0.0
1716425	NaN	NaN	0.0
1716426	0.0	0.0	0.0
1716427	0.0	NaN	0.0

	CREDIT_TYPE	DAYS_CREDIT_UPDATE	AMT_ANNUITY
0	Consumer credit	-131	NaN
1	Credit card	-20	NaN
2	Consumer credit	-16	NaN
3	Credit card	-16	NaN
4	Consumer credit	-21	NaN
...	...	...	...
1716423	Microloan	-19	NaN
1716424	Consumer credit	-2493	NaN
1716425	Consumer credit	-967	NaN
1716426	Consumer credit	-1508	NaN
1716427	Microloan	-387	NaN

[1716428 rows x 17 columns]

```
[ ]: bureau.columns
```

```
[ ]: Index(['SK_ID_CURR', 'SK_ID_BUREAU', 'CREDIT_ACTIVE', 'CREDIT_CURRENCY',
            'DAYS_CREDIT', 'CREDIT_DAY_OVERDUE', 'DAYS_CREDIT_ENDDATE',
            'DAYS_ENDDATE_FACT', 'AMT_CREDIT_MAX_OVERDUE', 'CNT_CREDIT_PROLONG',
            'AMT_CREDIT_SUM', 'AMT_CREDIT_SUM_DEBT', 'AMT_CREDIT_SUM_LIMIT',
            'AMT_CREDIT_SUM_OVERDUE', 'CREDIT_TYPE', 'DAYS_CREDIT_UPDATE',
            'AMT_ANNUITY'],
```

```
dtype='object')
```

```
[ ]: col_desc.loc[col_desc.Row.eq('SK_ID_CURR') & col_desc.Table.eq('bureau.csv')].  
      ↳Description.values
```

```
[ ]: array(['ID of loan in our sample - one loan in our sample can have 0,1,2 or more  
related previous credits in credit bureau '],  
          dtype=object)
```

```
[ ]: bureau.shape
```

```
[ ]: (1716428, 17)
```

```
[ ]: len(bureau.SK_ID_BUREAU.unique())
```

```
[ ]: 1716428
```

```
[ ]: len(bureau.SK_ID_CURR.unique())
```

```
[ ]: 305811
```

```
[ ]: bureau[bureau.duplicated(subset=['SK_ID_CURR']) == True]
```

```
[ ]: 
```

	SK_ID_CURR	SK_ID_BUREAU	CREDIT_ACTIVE	CREDIT_CURRENCY	DAYS_CREDIT	\
1	215354	5714463	Active	currency 1	-208	
2	215354	5714464	Active	currency 1	-203	
3	215354	5714465	Active	currency 1	-203	
4	215354	5714466	Active	currency 1	-629	
5	215354	5714467	Active	currency 1	-273	
...	...	...	...	...	...	
1716423	259355	5057750	Active	currency 1	-44	
1716424	100044	5057754	Closed	currency 1	-2648	
1716425	100044	5057762	Closed	currency 1	-1809	
1716426	246829	5057770	Closed	currency 1	-1878	
1716427	246829	5057778	Closed	currency 1	-463	

  

	CREDIT_DAY_OVERDUE	DAYS_CREDIT_ENDDATE	DAYS_ENDDATE_FACT	\
1	0	1075.0	NaN	
2	0	528.0	NaN	
3	0	NaN	NaN	
4	0	1197.0	NaN	
5	0	27460.0	NaN	
...	...	...	...	
1716423	0	-30.0	NaN	
1716424	0	-2433.0	-2493.0	
1716425	0	-1628.0	-970.0	
1716426	0	-1513.0	-1513.0	

1716427	0	NaN	-387.0
---------	---	-----	--------

	AMT_CREDIT_MAX_OVERDUE	CNT_CREDIT_PROLONG	AMT_CREDIT_SUM \
1	NaN	0	225000.00
2	NaN	0	464323.50
3	NaN	0	90000.00
4	77674.5	0	2700000.00
5	0.0	0	180000.00
...	...	...	...
1716423	0.0	0	11250.00
1716424	5476.5	0	38130.84
1716425	NaN	0	15570.00
1716426	NaN	0	36000.00
1716427	NaN	0	22500.00

	AMT_CREDIT_SUM_DEBT	AMT_CREDIT_SUM_LIMIT	AMT_CREDIT_SUM_OVERDUE \
1	171342.00	NaN	0.0
2	NaN	NaN	0.0
3	NaN	NaN	0.0
4	NaN	NaN	0.0
5	71017.38	108982.62	0.0
...	...	...	...
1716423	11250.00	0.00	0.0
1716424	0.00	0.00	0.0
1716425	NaN	NaN	0.0
1716426	0.00	0.00	0.0
1716427	0.00	NaN	0.0

	CREDIT_TYPE	DAYS_CREDIT_UPDATE	AMT_ANNUITY
1	Credit card	-20	NaN
2	Consumer credit	-16	NaN
3	Credit card	-16	NaN
4	Consumer credit	-21	NaN
5	Credit card	-31	NaN
...	...	...	...
1716423	Microloan	-19	NaN
1716424	Consumer credit	-2493	NaN
1716425	Consumer credit	-967	NaN
1716426	Consumer credit	-1508	NaN
1716427	Microloan	-387	NaN

[1410617 rows x 17 columns]