Home Credit Default Risk (HCDR)

The course project is based on the Home Credit Default Risk (HCDR) Kaggle Competition. The goal of this project is to predict whether or not a client will repay a loan. In order to make sure that people who struggle to get loans due to insufficient or non-existent credit histories have a positive loan experience, Home Credit makes use of a variety of alternative data--including telco and transactional information--to predict their clients' repayment abilities.

Some of the challenges

1. Dataset size
   - (688 meg uncompressed) with millions of rows of data
   - 2.71 Gig of data uncompressed
- Dealing with missing data
- Imbalanced datasets
- Summarizing transaction data

Kaggle API setup

Kaggle is a Data Science Competition Platform which shares a lot of datasets. In the past, it was troublesome to submit your result as your have to go through the console in your browser and drag your files there. Now you can interact with Kaggle via the command line. E.g.,

```
! kaggle competitions files home-credit-default-risk
```

It is quite easy to setup, it takes me less than 15 minutes to finish a submission.

1. Install library
- Create a API Token (edit your profile on Kaggle.com); this produces `kaggle.json` file
- Put your JSON `kaggle.json` in the right place
- Access competition files; make submissions via the command (see examples below)
- Submit result

For more detailed information on setting the Kaggle API see here and here.

```
!pip install kaggle
```

```
Requirement already satisfied: kaggle in
/usr/local/lib/python3.7/dist-packages (1.5.12)
Requirement already satisfied: python-slugify in
/usr/local/lib/python3.7/dist-packages (from kaggle) (6.1.1)
Requirement already satisfied: six>=1.10 in
/usr/local/lib/python3.7/dist-packages (from kaggle) (1.15.0)
```

```
Requirement already satisfied: requests in
/usr/local/lib/python3.7/dist-packages (from kaggle) (2.23.0)
Requirement already satisfied: python-dateutil in
/usr/local/lib/python3.7/dist-packages (from kaggle) (2.8.2)
Requirement already satisfied: certifi in
/usr/local/lib/python3.7/dist-packages (from kaggle) (2021.10.8)
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-
packages (from kaggle) (4.63.0)
Requirement already satisfied: urllib3 in
/usr/local/lib/python3.7/dist-packages (from kaggle) (1.24.3)
Requirement already satisfied: text-unidecode>=1.3 in
/usr/local/lib/python3.7/dist-packages (from python-slugify->kaggle)
(1.3)
Requirement already satisfied: idna<3,>=2.5 in
/usr/local/lib/python3.7/dist-packages (from requests->kaggle) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in
/usr/local/lib/python3.7/dist-packages (from requests->kaggle) (3.0.4)

!pwd
# /root/shared/PycharmProjects/I526_AML_Student/Assignments/Unit-
Project-Home-Credit-Default-Risk/WIP_phase-1/kaggle.json

/content


!ls

expLog.csv  home_credit.png  kaggle.json  sample_data

!rm -rf ~/.kaggle
!mkdir ~/.kaggle
!cp kaggle.json ~/.kaggle
!chmod 600 ~/.kaggle/kaggle.json

! kaggle competitions files home-credit-default-risk

name                                  size  creationDate
----------------------------------  -----  -------------------
credit_card_balance.csv             405MB  2019-12-11 02:55:35
bureau.csv                          162MB  2019-12-11 02:55:35
POS_CASH_balance.csv                375MB  2019-12-11 02:55:35
installments_payments.csv           690MB  2019-12-11 02:55:35
sample_submission.csv               524KB  2019-12-11 02:55:35
previous_application.csv            386MB  2019-12-11 02:55:35
bureau_balance.csv                  358MB  2019-12-11 02:55:35
application_train.csv               158MB  2019-12-11 02:55:35
HomeCredit_columns_description.csv   37KB  2019-12-11 02:55:35
application_test.csv                 25MB  2019-12-11 02:55:35
```

Dataset and how to download

Back ground Home Credit Group

Many people struggle to get loans due to insufficient or non-existent credit histories. And, unfortunately, this population is often taken advantage of by untrustworthy lenders.

Home Credit Group

Home Credit strives to broaden financial inclusion for the unbanked population by providing a positive and safe borrowing experience. In order to make sure this underserved population has a positive loan experience, Home Credit makes use of a variety of alternative data--including telco and transactional information--to predict their clients' repayment abilities.

While Home Credit is currently using various statistical and machine learning methods to make these predictions, they're challenging Kagglers to help them unlock the full potential of their data. Doing so will ensure that clients capable of repayment are not rejected and that loans are given with a principal, maturity, and repayment calendar that will empower their clients to be successful.

Background on the dataset

Home Credit is a non-banking financial institution, founded in 1997 in the Czech Republic.

The company operates in 14 countries (including United States, Russia, Kazahstan, Belarus, China, India) and focuses on lending primarily to people with little or no credit history which will either not obtain loans or became victims of untrustworthly lenders.

Home Credit group has over 29 million customers, total assests of 21 billions Euro, over 160 millions loans, with the majority in Asia and and almost half of them in China (as of 19-05-2018).

While Home Credit is currently using various statistical and machine learning methods to make these predictions, they're challenging Kagglers to help them unlock the full potential of their data. Doing so will ensure that clients capable of repayment are not rejected and that loans are given with a principal, maturity, and repayment calendar that will empower their clients to be successful.

Data files overview

There are 7 different sources of data:

- **application_train/application_test:** the main training and testing data with information about each loan application at Home Credit. Every loan has its own row and is identified by the feature SK_ID_CURR. The training application data comes with the TARGET indicating **0: the loan was repaid** or **1: the loan was not repaid**. The target variable defines if the client had payment difficulties meaning he/she had

late payment more than X days on at least one of the first Y installments of the loan. Such case is marked as 1 while other all other cases as 0.

- **bureau:** data concerning client's previous credits from other financial institutions. Each previous credit has its own row in bureau, but one loan in the application data can have multiple previous credits.
- **bureau_balance:** monthly data about the previous credits in bureau. Each row is one month of a previous credit, and a single previous credit can have multiple rows, one for each month of the credit length.
- **previous_application:** previous applications for loans at Home Credit of clients who have loans in the application data. Each current loan in the application data can have multiple previous loans. Each previous application has one row and is identified by the feature SK_ID_PREV.
- **POS_CASH_BALANCE:** monthly data about previous point of sale or cash loans clients have had with Home Credit. Each row is one month of a previous point of sale or cash loan, and a single previous loan can have many rows.
- credit_card_balance: monthly data about previous credit cards clients have had with Home Credit. Each row is one month of a credit card balance, and a single credit card can have many rows.
- **installments_payment:** payment history for previous loans at Home Credit. There is one row for every made payment and one row for every missed payment.

alt

home_credit.png

Downloading the files via Kaggle API

Create a base directory:

```
DATA_DIR = "../../../Data/home-credit-default-risk"   #same level as
course repo in the data directory
```

Please download the project data files and data dictionary and unzip them using either of the following approaches:

1. Click on the `Download` button on the following Data Webpage and unzip the zip file to the `BASE_DIR`
2. If you plan to use the Kaggle API, please use the following steps.

```
# Commented by kiran
# DATA_DIR = "../../../Data/home-credit-default-risk"   #same level as
course repo in the data directory
# DATA_DIR = os.path.join('./ddddd/')
# !mkdir $DATA_DIR

# data dir for kiran
DATA_DIR = "../Data/"
# Google collab dir: Account: kikarand@iu.edu
```

```
DATA_DIR="gdrive/MyDrive/data/"
!ls -l $DATA_DIR
```

```
total 2621352
-rw------- 1 root root  26567651 Apr 10 00:39 application_test.csv
-rw------- 1 root root 166133370 Apr 10 00:39 application_train.csv
-rw------- 1 root root 375592889 Apr 10 00:39 bureau_balance.csv
-rw------- 1 root root 170016717 Apr 10 00:39 bureau.csv
-rw------- 1 root root 424582605 Apr 10 00:39 credit_card_balance.csv
-rw------- 1 root root     37381 Apr 10 01:03
HomeCredit_columns_description.csv
-rw------- 1 root root 723118349 Apr 10 00:39
installments_payments.csv
-rw------- 1 root root 392703158 Apr 10 00:39 POS_CASH_balance.csv
-rw------- 1 root root 404973293 Apr 10 00:39 previous_application.csv
-rw------- 1 root root    536202 Apr 10 00:39 sample_submission.csv
```

Mount Gdrive

```
# Added to download files in google collab
from google.colab import drive,files
drive.mount('/content/gdrive')
#this will prompt you to upload the kaggle.json
# files.upload()
!ls -lha kaggle.json
```

```
Mounted at /content/gdrive
ls: cannot access 'kaggle.json': No such file or directory
```

```
!rm -rf ~/.kaggle
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
!chmod 600 /root/.kaggle/kaggle.json
DATA_DIR = "/Data/"
!mkdir $DATA_DIR
!ls -l $DATA_DIR

# IF download file do not exists
! kaggle competitions download -c home-credit-default-risk
```

```
403 - Forbidden
```

Imports

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import LabelEncoder
import os
import zipfile
from sklearn.base import BaseEstimator, TransformerMixin
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import MinMaxScaler
from sklearn.pipeline import Pipeline, FeatureUnion
from pandas.plotting import scatter_matrix
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import OneHotEncoder
import warnings
warnings.filterwarnings('ignore')
```

Extract Zip Files, ignore if unzipped already

```
# unzippingReq = False
# if unzippingReq: #please modify this code
#     zip_ref = zipfile.ZipFile('application_train.csv.zip', 'r')
#     zip_ref.extractall('datasets')
#     zip_ref.close()
#     zip_ref = zipfile.ZipFile('application_test.csv.zip', 'r')
#     zip_ref.extractall('datasets')
#     zip_ref.close()
#     zip_ref = zipfile.ZipFile('bureau_balance.csv.zip', 'r')
#     zip_ref.extractall('datasets')
#     zip_ref.close()
#     zip_ref = zipfile.ZipFile('bureau.csv.zip', 'r')
#     zip_ref.extractall('datasets')
#     zip_ref.close()
#     zip_ref = zipfile.ZipFile('credit_card_balance.csv.zip', 'r')
#     zip_ref.extractall('datasets')
#     zip_ref.close()
#     zip_ref = zipfile.ZipFile('installments_payments.csv.zip', 'r')
#     zip_ref.extractall('datasets')
#     zip_ref.close()
#     zip_ref = zipfile.ZipFile('POS_CASH_balance.csv.zip', 'r')
#     zip_ref.extractall('datasets')
#     zip_ref.close()
#     zip_ref = zipfile.ZipFile('previous_application.csv.zip', 'r')
#     zip_ref.extractall('datasets')
#     zip_ref.close()
```

Data files overview

Data Dictionary

As part of the data download comes a Data Dictionary. It named `HomeCredit_columns_description.csv`

```
data_dict_path = os.path.join(DATA_DIR,
"HomeCredit_columns_description.csv")
```

```
data_dict = pd.read_csv(data_dict_path, engine="python",
encoding="utf-8" ,header=0)

data_dict["Table"].unique()

array(['application_{train|test}.csv', 'bureau.csv',
'bureau_balance.csv',
       'POS_CASH_balance.csv', 'credit_card_balance.csv',
       'previous_application.csv', 'installments_payments.csv'],
      dtype=object)
```

image.png

Load Application train

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import LabelEncoder
import os
import zipfile
from sklearn.base import BaseEstimator, TransformerMixin
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import MinMaxScaler
from sklearn.pipeline import Pipeline, FeatureUnion
from pandas.plotting import scatter_matrix
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import OneHotEncoder
import warnings
warnings.filterwarnings('ignore')

def load_data(in_path, name):
    df = pd.read_csv(in_path)
    print(f"{name}: shape is {df.shape}")
    print(df.info())
    display(df.head(5))
    display(df.describe())
    display(df.isna().sum())
    return df

datasets={}  # lets store the datasets in a dictionary so we can keep
track of them easily
ds_name = 'application_train'
datasets[ds_name] = load_data(os.path.join(DATA_DIR,
f'{ds_name}.csv'), ds_name)
```

```
datasets['application_train'].shape
```

Application test

- **application_train/application_test:** the main training and testing data with information about each loan application at Home Credit. Every loan has its own row and is identified by the feature SK_ID_CURR. The training application data comes with the TARGET indicating **0: the loan was repaid** or **1: the loan was not repaid**. The target variable defines if the client had payment difficulties meaning he/she had late payment more than X days on at least one of the first Y installments of the loan. Such case is marked as 1 while other all other cases as 0.

```
ds_name = 'application_test'
datasets[ds_name] = load_data(os.path.join(DATA_DIR,
f'{ds_name}.csv'), ds_name)
```

```
application_test: shape is (48744, 121)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48744 entries, 0 to 48743
Columns: 121 entries, SK_ID_CURR to AMT_REQ_CREDIT_BUREAU_YEAR
dtypes: float64(65), int64(40), object(16)
memory usage: 45.0+ MB
None
```

```
   SK_ID_CURR NAME_CONTRACT_TYPE CODE_GENDER FLAG_OWN_CAR
FLAG_OWN_REALTY  \
0      100001         Cash loans           F            N
Y
1      100005         Cash loans           M            N
Y
2      100013         Cash loans           M            Y
Y
3      100028         Cash loans           F            N
Y
4      100038         Cash loans           M            Y
N

   CNT_CHILDREN  AMT_INCOME_TOTAL  AMT_CREDIT  AMT_ANNUITY
AMT_GOODS_PRICE  \
0             0          135000.0    568800.0      20560.5
450000.0
1             0           99000.0    222768.0      17370.0
180000.0
2             0          202500.0    663264.0      69777.0
630000.0
3             2          315000.0   1575000.0      49018.5
1575000.0
4             1          180000.0    625500.0      32067.0
625500.0
```

```
   ... 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                        0.0                        0.0
4                        NaN                        NaN

   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                         0.0                        0.0
4                         NaN                        NaN

   AMT_REQ_CREDIT_BUREAU_QRT  AMT_REQ_CREDIT_BUREAU_YEAR
0                        0.0                        0.0
1                        0.0                        3.0
2                        1.0                        4.0
3                        0.0                        3.0
4                        NaN                        NaN

[5 rows x 121 columns]

          SK_ID_CURR  CNT_CHILDREN  AMT_INCOME_TOTAL    AMT_CREDIT  \
count   48744.000000  48744.000000      4.874400e+04  4.874400e+04
mean   277796.676350      0.397054      1.784318e+05  5.167404e+05
std    103169.547296      0.709047      1.015226e+05  3.653970e+05
min    100001.000000      0.000000      2.694150e+04  4.500000e+04
25%    188557.750000      0.000000      1.125000e+05  2.606400e+05
50%    277549.000000      0.000000      1.575000e+05  4.500000e+05
75%    367555.500000      1.000000      2.250000e+05  6.750000e+05
max    456250.000000     20.000000      4.410000e+06  2.245500e+06

         AMT_ANNUITY  AMT_GOODS_PRICE  REGION_POPULATION_RELATIVE  \
count   48720.000000     4.874400e+04                48744.000000
mean    29426.240209     4.626188e+05                    0.021226
std     16016.368315     3.367102e+05                    0.014428
```

```
min      2295.000000     4.500000e+04                    0.000253
25%     17973.000000     2.250000e+05                    0.010006
50%     26199.000000     3.960000e+05                    0.018850
75%     37390.500000     6.300000e+05                    0.028663
max    180576.000000     2.245500e+06                    0.072508

         DAYS_BIRTH  DAYS_EMPLOYED  DAYS_REGISTRATION  ...
FLAG_DOCUMENT_18  \
count  48744.000000   48744.000000       48744.000000  ...
48744.000000
mean  -16068.084605   67485.366322       -4967.652716  ...
0.001559
std     4325.900393  144348.507136        3552.612035  ...
0.039456
min   -25195.000000  -17463.000000      -23722.000000  ...
0.000000
25%   -19637.000000   -2910.000000       -7459.250000  ...
0.000000
50%   -15785.000000   -1293.000000       -4490.000000  ...
0.000000
75%   -12496.000000    -296.000000       -1901.000000  ...
0.000000
max    -7338.000000  365243.000000           0.000000  ...
1.000000

       FLAG_DOCUMENT_19  FLAG_DOCUMENT_20  FLAG_DOCUMENT_21  \
count           48744.0           48744.0           48744.0
mean                0.0               0.0               0.0
std                 0.0               0.0               0.0
min                 0.0               0.0               0.0
25%                 0.0               0.0               0.0
50%                 0.0               0.0               0.0
75%                 0.0               0.0               0.0
max                 0.0               0.0               0.0

       AMT_REQ_CREDIT_BUREAU_HOUR  AMT_REQ_CREDIT_BUREAU_DAY  \
count                42695.000000               42695.000000
mean                     0.002108                   0.001803
std                      0.046373                   0.046132
min                      0.000000                   0.000000
25%                      0.000000                   0.000000
50%                      0.000000                   0.000000
75%                      0.000000                   0.000000
max                      2.000000                   2.000000

       AMT_REQ_CREDIT_BUREAU_WEEK  AMT_REQ_CREDIT_BUREAU_MON  \
count                42695.000000               42695.000000
mean                     0.002787                   0.009299
std                      0.054037                   0.110924
min                      0.000000                   0.000000
```

```
25%                          0.000000                     0.000000
50%                          0.000000                     0.000000
75%                          0.000000                     0.000000
max                          2.000000                     6.000000

       AMT_REQ_CREDIT_BUREAU_QRT  AMT_REQ_CREDIT_BUREAU_YEAR
count                42695.000000                42695.000000
mean                     0.546902                    1.983769
std                      0.693305                    1.838873
min                      0.000000                    0.000000
25%                      0.000000                    0.000000
50%                      0.000000                    2.000000
75%                      1.000000                    3.000000
max                      7.000000                   17.000000

[8 rows x 105 columns]
```

```
SK_ID_CURR                      0
NAME_CONTRACT_TYPE              0
CODE_GENDER                     0
FLAG_OWN_CAR                    0
FLAG_OWN_REALTY                 0
                               ...
AMT_REQ_CREDIT_BUREAU_DAY    6049
AMT_REQ_CREDIT_BUREAU_WEEK   6049
AMT_REQ_CREDIT_BUREAU_MON    6049
AMT_REQ_CREDIT_BUREAU_QRT    6049
AMT_REQ_CREDIT_BUREAU_YEAR   6049
Length: 121, dtype: int64
```

The application dataset has the most information about the client: Gender, income, family status, education ...

The Other datasets

- **bureau:** data concerning client's previous credits from other financial institutions. Each previous credit has its own row in bureau, but one loan in the application data can have multiple previous credits.
- **bureau_balance:** monthly data about the previous credits in bureau. Each row is one month of a previous credit, and a single previous credit can have multiple rows, one for each month of the credit length.
- **previous_application:** previous applications for loans at Home Credit of clients who have loans in the application data. Each current loan in the application data can have multiple previous loans. Each previous application has one row and is identified by the feature SK_ID_PREV.
- **POS_CASH_BALANCE:** monthly data about previous point of sale or cash loans clients have had with Home Credit. Each row is one month of a previous point of sale or cash loan, and a single previous loan can have many rows.

- credit_card_balance: monthly data about previous credit cards clients have had with Home Credit. Each row is one month of a credit card balance, and a single credit card can have many rows.
- **installments_payment:** payment history for previous loans at Home Credit. There is one row for every made payment and one row for every missed payment.

```
%%time
def load_data_files():
  ds_names = ("application_train", "application_test",
"bureau","bureau_balance","credit_card_balance","installments_payments
",
            "previous_application","POS_CASH_balance")

  for ds_name in ds_names:
      datasets[ds_name] = load_data(os.path.join(DATA_DIR,
f'{ds_name}.csv'), ds_name)
```

```
application_train: shape is (307511, 122)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Columns: 122 entries, SK_ID_CURR to AMT_REQ_CREDIT_BUREAU_YEAR
dtypes: float64(65), int64(41), object(16)
memory usage: 286.2+ MB
None

   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]

          SK_ID_CURR         TARGET   CNT_CHILDREN
AMT_INCOME_TOTAL  \
count  307511.000000  307511.000000  307511.000000      3.075110e+05

mean   278180.518577       0.080729       0.417052      1.687979e+05

std    102790.175348       0.272419       0.722121      2.371231e+05

min    100002.000000       0.000000       0.000000      2.565000e+04

25%    189145.500000       0.000000       0.000000      1.125000e+05

50%    278202.000000       0.000000       0.000000      1.471500e+05

75%    367142.500000       0.000000       1.000000      2.025000e+05

max    456255.000000       1.000000      19.000000      1.170000e+08
```

```
         AMT_CREDIT    AMT_ANNUITY  AMT_GOODS_PRICE  \
count  3.075110e+05  307499.000000     3.072330e+05
mean   5.990260e+05   27108.573909     5.383962e+05
std    4.024908e+05   14493.737315     3.694465e+05
min    4.500000e+04    1615.500000     4.050000e+04
25%    2.700000e+05   16524.000000     2.385000e+05
50%    5.135310e+05   24903.000000     4.500000e+05
75%    8.086500e+05   34596.000000     6.795000e+05
max    4.050000e+06  258025.500000     4.050000e+06

       REGION_POPULATION_RELATIVE     DAYS_BIRTH
DAYS_EMPLOYED  ...  \
count               307511.000000  307511.000000  307511.000000  ...

mean                     0.020868  -16036.995067   63815.045904  ...

std                      0.013831    4363.988632  141275.766519  ...

min                      0.000290  -25229.000000  -17912.000000  ...

25%                      0.010006  -19682.000000   -2760.000000  ...

50%                      0.018850  -15750.000000   -1213.000000  ...

75%                      0.028663  -12413.000000    -289.000000  ...

max                      0.072508   -7489.000000  365243.000000  ...


       FLAG_DOCUMENT_18  FLAG_DOCUMENT_19  FLAG_DOCUMENT_20
FLAG_DOCUMENT_21  \
count     307511.000000     307511.000000     307511.000000
307511.000000
mean           0.008130          0.000595          0.000507
0.000335
std            0.089798          0.024387          0.022518
0.018299
min            0.000000          0.000000          0.000000
0.000000
25%            0.000000          0.000000          0.000000
0.000000
50%            0.000000          0.000000          0.000000
0.000000
75%            0.000000          0.000000          0.000000
0.000000
max            1.000000          1.000000          1.000000
1.000000

       AMT_REQ_CREDIT_BUREAU_HOUR  AMT_REQ_CREDIT_BUREAU_DAY  \
```

```
count                265992.000000                265992.000000
mean                      0.006402                     0.007000
std                       0.083849                     0.110757
min                       0.000000                     0.000000
25%                       0.000000                     0.000000
50%                       0.000000                     0.000000
75%                       0.000000                     0.000000
max                       4.000000                     9.000000

       AMT_REQ_CREDIT_BUREAU_WEEK  AMT_REQ_CREDIT_BUREAU_MON  \
count                265992.000000              265992.000000
mean                      0.034362                   0.267395
std                       0.204685                   0.916002
min                       0.000000                   0.000000
25%                       0.000000                   0.000000
50%                       0.000000                   0.000000
75%                       0.000000                   0.000000
max                       8.000000                  27.000000

       AMT_REQ_CREDIT_BUREAU_QRT  AMT_REQ_CREDIT_BUREAU_YEAR
count              265992.000000               265992.000000
mean                    0.265474                    1.899974
std                     0.794056                    1.869295
min                     0.000000                    0.000000
25%                     0.000000                    0.000000
50%                     0.000000                    1.000000
75%                     0.000000                    3.000000
max                   261.000000                   25.000000

[8 rows x 106 columns]

SK_ID_CURR                      0
TARGET                          0
NAME_CONTRACT_TYPE              0
CODE_GENDER                     0
FLAG_OWN_CAR                    0
                             ...
AMT_REQ_CREDIT_BUREAU_DAY     41519
AMT_REQ_CREDIT_BUREAU_WEEK    41519
AMT_REQ_CREDIT_BUREAU_MON     41519
AMT_REQ_CREDIT_BUREAU_QRT     41519
AMT_REQ_CREDIT_BUREAU_YEAR    41519
Length: 122, dtype: int64

application_test: shape is (48744, 121)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48744 entries, 0 to 48743
Columns: 121 entries, SK_ID_CURR to AMT_REQ_CREDIT_BUREAU_YEAR
dtypes: float64(65), int64(40), object(16)
```

```
memory usage: 45.0+ MB
None

   SK_ID_CURR NAME_CONTRACT_TYPE CODE_GENDER FLAG_OWN_CAR
FLAG_OWN_REALTY  \
0      100001         Cash loans           F            N
Y
1      100005         Cash loans           M            N
Y
2      100013         Cash loans           M            Y
Y
3      100028         Cash loans           F            N
Y
4      100038         Cash loans           M            Y
N

   CNT_CHILDREN  AMT_INCOME_TOTAL  AMT_CREDIT  AMT_ANNUITY
AMT_GOODS_PRICE  \
0             0          135000.0    568800.0      20560.5
450000.0
1             0           99000.0    222768.0      17370.0
180000.0
2             0          202500.0    663264.0      69777.0
630000.0
3             2          315000.0   1575000.0      49018.5
1575000.0
4             1          180000.0    625500.0      32067.0
625500.0

   ... 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                        0.0                         0.0
4                        NaN                         NaN

   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                           0.0                           0.0
4                           NaN                           NaN

   AMT_REQ_CREDIT_BUREAU_QRT  AMT_REQ_CREDIT_BUREAU_YEAR
0                        0.0                         0.0
1                        0.0                         3.0
2                        1.0                         4.0
3                        0.0                         3.0
4                        NaN                         NaN

[5 rows x 121 columns]

          SK_ID_CURR  CNT_CHILDREN  AMT_INCOME_TOTAL    AMT_CREDIT  \
count   48744.000000  48744.000000      4.874400e+04  4.874400e+04
mean   277796.676350      0.397054      1.784318e+05  5.167404e+05
std    103169.547296      0.709047      1.015226e+05  3.653970e+05
min    100001.000000      0.000000      2.694150e+04  4.500000e+04
25%    188557.750000      0.000000      1.125000e+05  2.606400e+05
50%    277549.000000      0.000000      1.575000e+05  4.500000e+05
75%    367555.500000      1.000000      2.250000e+05  6.750000e+05
max    456250.000000     20.000000      4.410000e+06  2.245500e+06

         AMT_ANNUITY  AMT_GOODS_PRICE  REGION_POPULATION_RELATIVE  \
count   48720.000000     4.874400e+04                48744.000000
mean    29426.240209     4.626188e+05                    0.021226
std     16016.368315     3.367102e+05                    0.014428
min      2295.000000     4.500000e+04                    0.000253
25%     17973.000000     2.250000e+05                    0.010006
50%     26199.000000     3.960000e+05                    0.018850
75%     37390.500000     6.300000e+05                    0.028663
max    180576.000000     2.245500e+06                    0.072508

         DAYS_BIRTH  DAYS_EMPLOYED  DAYS_REGISTRATION  ...
FLAG_DOCUMENT_18  \
count  48744.000000   48744.000000       48744.000000  ...
48744.000000
mean  -16068.084605   67485.366322       -4967.652716  ...
0.001559
std     4325.900393  144348.507136        3552.612035  ...
0.039456
min   -25195.000000  -17463.000000      -23722.000000  ...
0.000000
25%   -19637.000000   -2910.000000       -7459.250000  ...
0.000000
50%   -15785.000000   -1293.000000       -4490.000000  ...
0.000000
75%   -12496.000000    -296.000000       -1901.000000  ...
```

```
0.000000
max    -7338.000000  365243.000000           0.000000  ...
1.000000

       FLAG_DOCUMENT_19  FLAG_DOCUMENT_20  FLAG_DOCUMENT_21  \
count           48744.0           48744.0           48744.0
mean                0.0               0.0               0.0
std                 0.0               0.0               0.0
min                 0.0               0.0               0.0
25%                 0.0               0.0               0.0
50%                 0.0               0.0               0.0
75%                 0.0               0.0               0.0
max                 0.0               0.0               0.0

       AMT_REQ_CREDIT_BUREAU_HOUR  AMT_REQ_CREDIT_BUREAU_DAY  \
count                42695.000000               42695.000000
mean                     0.002108                   0.001803
std                      0.046373                   0.046132
min                      0.000000                   0.000000
25%                      0.000000                   0.000000
50%                      0.000000                   0.000000
75%                      0.000000                   0.000000
max                      2.000000                   2.000000

       AMT_REQ_CREDIT_BUREAU_WEEK  AMT_REQ_CREDIT_BUREAU_MON  \
count                42695.000000               42695.000000
mean                     0.002787                   0.009299
std                      0.054037                   0.110924
min                      0.000000                   0.000000
25%                      0.000000                   0.000000
50%                      0.000000                   0.000000
75%                      0.000000                   0.000000
max                      2.000000                   6.000000

       AMT_REQ_CREDIT_BUREAU_QRT  AMT_REQ_CREDIT_BUREAU_YEAR
count                42695.000000               42695.000000
mean                     0.546902                   1.983769
std                      0.693305                   1.838873
min                      0.000000                   0.000000
25%                      0.000000                   0.000000
50%                      0.000000                   2.000000
75%                      1.000000                   3.000000
max                      7.000000                  17.000000

[8 rows x 105 columns]

SK_ID_CURR                      0
NAME_CONTRACT_TYPE              0
CODE_GENDER                     0
FLAG_OWN_CAR                    0
```

```
FLAG_OWN_REALTY                 0
                              ...
AMT_REQ_CREDIT_BUREAU_DAY     6049
AMT_REQ_CREDIT_BUREAU_WEEK    6049
AMT_REQ_CREDIT_BUREAU_MON     6049
AMT_REQ_CREDIT_BUREAU_QRT     6049
AMT_REQ_CREDIT_BUREAU_YEAR    6049
Length: 121, dtype: int64

bureau: shape is (1716428, 17)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1716428 entries, 0 to 1716427
Data columns (total 17 columns):
 #   Column                  Dtype
---  ------                  -----
 0   SK_ID_CURR              int64
 1   SK_ID_BUREAU            int64
 2   CREDIT_ACTIVE           object
 3   CREDIT_CURRENCY         object
 4   DAYS_CREDIT             int64
 5   CREDIT_DAY_OVERDUE      int64
 6   DAYS_CREDIT_ENDDATE     float64
 7   DAYS_ENDDATE_FACT       float64
 8   AMT_CREDIT_MAX_OVERDUE  float64
 9   CNT_CREDIT_PROLONG      int64
 10  AMT_CREDIT_SUM          float64
 11  AMT_CREDIT_SUM_DEBT     float64
 12  AMT_CREDIT_SUM_LIMIT    float64
 13  AMT_CREDIT_SUM_OVERDUE  float64
 14  CREDIT_TYPE             object
 15  DAYS_CREDIT_UPDATE      int64
 16  AMT_ANNUITY             float64
dtypes: float64(8), int64(6), object(3)
memory usage: 222.6+ MB
None

   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


   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

   AMT_CREDIT_MAX_OVERDUE  CNT_CREDIT_PROLONG  AMT_CREDIT_SUM  \
0                     NaN                   0         91323.0
1                     NaN                   0        225000.0
2                     NaN                   0        464323.5
3                     NaN                   0         90000.0
4                 77674.5                   0       2700000.0

   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


       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

         SK_ID_CURR  SK_ID_BUREAU   DAYS_CREDIT  CREDIT_DAY_OVERDUE  \
count  1.716428e+06  1.716428e+06  1.716428e+06        1.716428e+06
mean   2.782149e+05  5.924434e+06 -1.142108e+03        8.181666e-01
std    1.029386e+05  5.322657e+05  7.951649e+02        3.654443e+01
min    1.000010e+05  5.000000e+06 -2.922000e+03        0.000000e+00
25%    1.888668e+05  5.463954e+06 -1.666000e+03        0.000000e+00
50%    2.780550e+05  5.926304e+06 -9.870000e+02        0.000000e+00
75%    3.674260e+05  6.385681e+06 -4.740000e+02        0.000000e+00
max    4.562550e+05  6.843457e+06  0.000000e+00        2.792000e+03

       DAYS_CREDIT_ENDDATE  DAYS_ENDDATE_FACT  AMT_CREDIT_MAX_OVERDUE
\
count         1.610875e+06       1.082775e+06            5.919400e+05

mean          5.105174e+02      -1.017437e+03            3.825418e+03
```

```
std           4.994220e+03       7.140106e+02            2.060316e+05

min          -4.206000e+04      -4.202300e+04            0.000000e+00

25%          -1.138000e+03      -1.489000e+03            0.000000e+00

50%          -3.300000e+02      -8.970000e+02            0.000000e+00

75%           4.740000e+02      -4.250000e+02            0.000000e+00

max           3.119900e+04       0.000000e+00            1.159872e+08


       CNT_CREDIT_PROLONG  AMT_CREDIT_SUM  AMT_CREDIT_SUM_DEBT  \
count        1.716428e+06    1.716415e+06         1.458759e+06
mean         6.410406e-03    3.549946e+05         1.370851e+05
std          9.622391e-02    1.149811e+06         6.774011e+05
min          0.000000e+00    0.000000e+00        -4.705600e+06
25%          0.000000e+00    5.130000e+04         0.000000e+00
50%          0.000000e+00    1.255185e+05         0.000000e+00
75%          0.000000e+00    3.150000e+05         4.015350e+04
max          9.000000e+00    5.850000e+08         1.701000e+08

       AMT_CREDIT_SUM_LIMIT  AMT_CREDIT_SUM_OVERDUE
DAYS_CREDIT_UPDATE  \
count          1.124648e+06            1.716428e+06
1.716428e+06
mean           6.229515e+03            3.791276e+01       -
5.937483e+02
std            4.503203e+04            5.937650e+03
7.207473e+02
min           -5.864061e+05            0.000000e+00       -
4.194700e+04
25%            0.000000e+00            0.000000e+00       -
9.080000e+02
50%            0.000000e+00            0.000000e+00       -
3.950000e+02
75%            0.000000e+00            0.000000e+00       -
3.300000e+01
max            4.705600e+06            3.756681e+06
3.720000e+02

        AMT_ANNUITY
count  4.896370e+05
mean   1.571276e+04
std    3.258269e+05
min    0.000000e+00
25%    0.000000e+00
50%    0.000000e+00
```

```
75%    1.350000e+04
max    1.184534e+08

SK_ID_CURR                      0
SK_ID_BUREAU                    0
CREDIT_ACTIVE                   0
CREDIT_CURRENCY                 0
DAYS_CREDIT                     0
CREDIT_DAY_OVERDUE              0
DAYS_CREDIT_ENDDATE        105553
DAYS_ENDDATE_FACT          633653
AMT_CREDIT_MAX_OVERDUE    1124488
CNT_CREDIT_PROLONG              0
AMT_CREDIT_SUM                 13
AMT_CREDIT_SUM_DEBT        257669
AMT_CREDIT_SUM_LIMIT       591780
AMT_CREDIT_SUM_OVERDUE          0
CREDIT_TYPE                     0
DAYS_CREDIT_UPDATE              0
AMT_ANNUITY               1226791
dtype: int64

bureau_balance: shape is (27299925, 3)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27299925 entries, 0 to 27299924
Data columns (total 3 columns):
 #   Column          Dtype
---  ------          -----
 0   SK_ID_BUREAU    int64
 1   MONTHS_BALANCE  int64
 2   STATUS          object
dtypes: int64(2), object(1)
memory usage: 624.8+ MB
None

   SK_ID_BUREAU  MONTHS_BALANCE STATUS
0       5715448               0      C
1       5715448              -1      C
2       5715448              -2      C
3       5715448              -3      C
4       5715448              -4      C

       SK_ID_BUREAU  MONTHS_BALANCE
count  2.729992e+07    2.729992e+07
mean   6.036297e+06   -3.074169e+01
std    4.923489e+05    2.386451e+01
min    5.001709e+06   -9.600000e+01
25%    5.730933e+06   -4.600000e+01
50%    6.070821e+06   -2.500000e+01
75%    6.431951e+06   -1.100000e+01
max    6.842888e+06    0.000000e+00
```

```
SK_ID_BUREAU      0
MONTHS_BALANCE    0
STATUS            0
dtype: int64

credit_card_balance: shape is (3840312, 23)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3840312 entries, 0 to 3840311
Data columns (total 23 columns):
 #   Column                       Dtype
---  ------                       -----
 0   SK_ID_PREV                   int64
 1   SK_ID_CURR                   int64
 2   MONTHS_BALANCE               int64
 3   AMT_BALANCE                  float64
 4   AMT_CREDIT_LIMIT_ACTUAL      int64
 5   AMT_DRAWINGS_ATM_CURRENT     float64
 6   AMT_DRAWINGS_CURRENT         float64
 7   AMT_DRAWINGS_OTHER_CURRENT   float64
 8   AMT_DRAWINGS_POS_CURRENT     float64
 9   AMT_INST_MIN_REGULARITY      float64
 10  AMT_PAYMENT_CURRENT          float64
 11  AMT_PAYMENT_TOTAL_CURRENT    float64
 12  AMT_RECEIVABLE_PRINCIPAL     float64
 13  AMT_RECIVABLE                float64
 14  AMT_TOTAL_RECEIVABLE         float64
 15  CNT_DRAWINGS_ATM_CURRENT     float64
 16  CNT_DRAWINGS_CURRENT         int64
 17  CNT_DRAWINGS_OTHER_CURRENT   float64
 18  CNT_DRAWINGS_POS_CURRENT     float64
 19  CNT_INSTALMENT_MATURE_CUM    float64
 20  NAME_CONTRACT_STATUS         object
 21  SK_DPD                       int64
 22  SK_DPD_DEF                   int64
dtypes: float64(15), int64(7), object(1)
memory usage: 673.9+ MB
None

   SK_ID_PREV  SK_ID_CURR  MONTHS_BALANCE  AMT_BALANCE  \
0     2562384      378907              -6       56.970
1     2582071      363914              -1    63975.555
2     1740877      371185              -7    31815.225
3     1389973      337855              -4   236572.110
4     1891521      126868              -1   453919.455

   AMT_CREDIT_LIMIT_ACTUAL  AMT_DRAWINGS_ATM_CURRENT
AMT_DRAWINGS_CURRENT  \
0                   135000                      0.0
877.5
1                    45000                   2250.0
```

```
2250.0
2                    450000                         0.0
0.0
3                    225000                      2250.0
2250.0
4                    450000                         0.0
11547.0

   AMT_DRAWINGS_OTHER_CURRENT  AMT_DRAWINGS_POS_CURRENT  \
0                         0.0                     877.5
1                         0.0                       0.0
2                         0.0                       0.0
3                         0.0                       0.0
4                         0.0                   11547.0

   AMT_INST_MIN_REGULARITY  ...  AMT_RECIVABLE
AMT_TOTAL_RECEIVABLE  \
0                 1700.325  ...          0.000                 0.000

1                 2250.000  ...      64875.555             64875.555

2                 2250.000  ...      31460.085             31460.085

3                11795.760  ...     233048.970            233048.970

4                22924.890  ...     453919.455            453919.455


   CNT_DRAWINGS_ATM_CURRENT  CNT_DRAWINGS_CURRENT
CNT_DRAWINGS_OTHER_CURRENT  \
0                       0.0                     1
0.0
1                       1.0                     1
0.0
2                       0.0                     0
0.0
3                       1.0                     1
0.0
4                       0.0                     1
0.0

   CNT_DRAWINGS_POS_CURRENT  CNT_INSTALMENT_MATURE_CUM
NAME_CONTRACT_STATUS  \
0                       1.0                      35.0
Active
1                       0.0                      69.0
Active
2                       0.0                      30.0
Active
```

```
3                       0.0                        10.0
Active
4                       1.0                       101.0
Active

   SK_DPD  SK_DPD_DEF
0       0           0
1       0           0
2       0           0
3       0           0
4       0           0

[5 rows x 23 columns]

         SK_ID_PREV    SK_ID_CURR  MONTHS_BALANCE   AMT_BALANCE  \
count  3.840312e+06  3.840312e+06    3.840312e+06  3.840312e+06
mean   1.904504e+06  2.783242e+05   -3.452192e+01  5.830016e+04
std    5.364695e+05  1.027045e+05    2.666775e+01  1.063070e+05
min    1.000018e+06  1.000060e+05   -9.600000e+01 -4.202502e+05
25%    1.434385e+06  1.895170e+05   -5.500000e+01  0.000000e+00
50%    1.897122e+06  2.783960e+05   -2.800000e+01  0.000000e+00
75%    2.369328e+06  3.675800e+05   -1.100000e+01  8.904669e+04
max    2.843496e+06  4.562500e+05   -1.000000e+00  1.505902e+06

       AMT_CREDIT_LIMIT_ACTUAL  AMT_DRAWINGS_ATM_CURRENT  \
count             3.840312e+06              3.090496e+06
mean              1.538080e+05              5.961325e+03
std               1.651457e+05              2.822569e+04
min               0.000000e+00             -6.827310e+03
25%               4.500000e+04              0.000000e+00
50%               1.125000e+05              0.000000e+00
75%               1.800000e+05              0.000000e+00
max               1.350000e+06              2.115000e+06

       AMT_DRAWINGS_CURRENT  AMT_DRAWINGS_OTHER_CURRENT  \
count          3.840312e+06                3.090496e+06
mean           7.433388e+03                2.881696e+02
std            3.384608e+04                8.201989e+03
min           -6.211620e+03                0.000000e+00
25%            0.000000e+00                0.000000e+00
50%            0.000000e+00                0.000000e+00
75%            0.000000e+00                0.000000e+00
max            2.287098e+06                1.529847e+06

       AMT_DRAWINGS_POS_CURRENT  AMT_INST_MIN_REGULARITY  ...  \
count              3.090496e+06             3.535076e+06  ...
mean               2.968805e+03             3.540204e+03  ...
std                2.079689e+04             5.600154e+03  ...
min                0.000000e+00             0.000000e+00  ...
25%                0.000000e+00             0.000000e+00  ...
```

```
50%                0.000000e+00             0.000000e+00  ...
75%                0.000000e+00             6.633911e+03  ...
max                2.239274e+06             2.028820e+05  ...

       AMT_RECEIVABLE_PRINCIPAL  AMT_RECIVABLE
AMT_TOTAL_RECEIVABLE  \
count              3.840312e+06   3.840312e+06          3.840312e+06

mean               5.596588e+04   5.808881e+04          5.809829e+04

std                1.025336e+05   1.059654e+05          1.059718e+05

min               -4.233058e+05  -4.202502e+05         -4.202502e+05

25%                0.000000e+00   0.000000e+00          0.000000e+00

50%                0.000000e+00   0.000000e+00          0.000000e+00

75%                8.535924e+04   8.889949e+04          8.891451e+04

max                1.472317e+06   1.493338e+06          1.493338e+06


       CNT_DRAWINGS_ATM_CURRENT  CNT_DRAWINGS_CURRENT  \
count              3.090496e+06          3.840312e+06
mean               3.094490e-01          7.031439e-01
std                1.100401e+00          3.190347e+00
min                0.000000e+00          0.000000e+00
25%                0.000000e+00          0.000000e+00
50%                0.000000e+00          0.000000e+00
75%                0.000000e+00          0.000000e+00
max                5.100000e+01          1.650000e+02

       CNT_DRAWINGS_OTHER_CURRENT  CNT_DRAWINGS_POS_CURRENT  \
count                3.090496e+06              3.090496e+06
mean                 4.812496e-03              5.594791e-01
std                  8.263861e-02              3.240649e+00
min                  0.000000e+00              0.000000e+00
25%                  0.000000e+00              0.000000e+00
50%                  0.000000e+00              0.000000e+00
75%                  0.000000e+00              0.000000e+00
max                  1.200000e+01              1.650000e+02

       CNT_INSTALMENT_MATURE_CUM        SK_DPD    SK_DPD_DEF
count               3.535076e+06  3.840312e+06  3.840312e+06
mean                2.082508e+01  9.283667e+00  3.316220e-01
std                 2.005149e+01  9.751570e+01  2.147923e+01
min                 0.000000e+00  0.000000e+00  0.000000e+00
25%                 4.000000e+00  0.000000e+00  0.000000e+00
```

```
50%                  1.500000e+01  0.000000e+00  0.000000e+00
75%                  3.200000e+01  0.000000e+00  0.000000e+00
max                  1.200000e+02  3.260000e+03  3.260000e+03

[8 rows x 22 columns]

SK_ID_PREV                         0
SK_ID_CURR                         0
MONTHS_BALANCE                     0
AMT_BALANCE                        0
AMT_CREDIT_LIMIT_ACTUAL            0
AMT_DRAWINGS_ATM_CURRENT      749816
AMT_DRAWINGS_CURRENT               0
AMT_DRAWINGS_OTHER_CURRENT    749816
AMT_DRAWINGS_POS_CURRENT      749816
AMT_INST_MIN_REGULARITY       305236
AMT_PAYMENT_CURRENT           767988
AMT_PAYMENT_TOTAL_CURRENT          0
AMT_RECEIVABLE_PRINCIPAL           0
AMT_RECIVABLE                      0
AMT_TOTAL_RECEIVABLE               0
CNT_DRAWINGS_ATM_CURRENT      749816
CNT_DRAWINGS_CURRENT               0
CNT_DRAWINGS_OTHER_CURRENT    749816
CNT_DRAWINGS_POS_CURRENT      749816
CNT_INSTALMENT_MATURE_CUM     305236
NAME_CONTRACT_STATUS               0
SK_DPD                             0
SK_DPD_DEF                         0
dtype: int64

installments_payments: shape is (13605401, 8)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13605401 entries, 0 to 13605400
Data columns (total 8 columns):
 #   Column                  Dtype
---  ------                  -----
 0   SK_ID_PREV              int64
 1   SK_ID_CURR              int64
 2   NUM_INSTALMENT_VERSION  float64
 3   NUM_INSTALMENT_NUMBER   int64
 4   DAYS_INSTALMENT         float64
 5   DAYS_ENTRY_PAYMENT      float64
 6   AMT_INSTALMENT          float64
 7   AMT_PAYMENT             float64
dtypes: float64(5), int64(3)
memory usage: 830.4 MB
None

   SK_ID_PREV  SK_ID_CURR  NUM_INSTALMENT_VERSION
NUM_INSTALMENT_NUMBER  \
```

```
0     1054186      161674                    1.0
6
1     1330831      151639                    0.0
34
2     2085231      193053                    2.0
1
3     2452527      199697                    1.0
3
4     2714724      167756                    1.0
2
```

	DAYS_INSTALMENT	DAYS_ENTRY_PAYMENT	AMT_INSTALMENT	AMT_PAYMENT
0	-1180.0	-1187.0	6948.360	6948.360
1	-2156.0	-2156.0	1716.525	1716.525
2	-63.0	-63.0	25425.000	25425.000
3	-2418.0	-2426.0	24350.130	24350.130
4	-1383.0	-1366.0	2165.040	2160.585

	SK_ID_PREV	SK_ID_CURR	NUM_INSTALMENT_VERSION	\
count	1.360540e+07	1.360540e+07	1.360540e+07	
mean	1.903365e+06	2.784449e+05	8.566373e-01	
std	5.362029e+05	1.027183e+05	1.035216e+00	
min	1.000001e+06	1.000010e+05	0.000000e+00	
25%	1.434191e+06	1.896390e+05	0.000000e+00	
50%	1.896520e+06	2.786850e+05	1.000000e+00	
75%	2.369094e+06	3.675300e+05	1.000000e+00	
max	2.843499e+06	4.562550e+05	1.780000e+02	

	NUM_INSTALMENT_NUMBER	DAYS_INSTALMENT	DAYS_ENTRY_PAYMENT	\
count	1.360540e+07	1.360540e+07	1.360250e+07	
mean	1.887090e+01	-1.042270e+03	-1.051114e+03	
std	2.666407e+01	8.009463e+02	8.005859e+02	
min	1.000000e+00	-2.922000e+03	-4.921000e+03	
25%	4.000000e+00	-1.654000e+03	-1.662000e+03	
50%	8.000000e+00	-8.180000e+02	-8.270000e+02	
75%	1.900000e+01	-3.610000e+02	-3.700000e+02	
max	2.770000e+02	-1.000000e+00	-1.000000e+00	

	AMT_INSTALMENT	AMT_PAYMENT
count	1.360540e+07	1.360250e+07
mean	1.705091e+04	1.723822e+04
std	5.057025e+04	5.473578e+04
min	0.000000e+00	0.000000e+00
25%	4.226085e+03	3.398265e+03
50%	8.884080e+03	8.125515e+03
75%	1.671021e+04	1.610842e+04
max	3.771488e+06	3.771488e+06

SK_ID_PREV	0
SK_ID_CURR	0

```
NUM_INSTALMENT_VERSION       0
NUM_INSTALMENT_NUMBER        0
DAYS_INSTALMENT              0
DAYS_ENTRY_PAYMENT        2905
AMT_INSTALMENT               0
AMT_PAYMENT               2905
dtype: int64

previous_application: shape is (1670214, 37)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1670214 entries, 0 to 1670213
Data columns (total 37 columns):
 #   Column                       Non-Null Count    Dtype
---  ------                       --------------    -----
 0   SK_ID_PREV                   1670214 non-null  int64
 1   SK_ID_CURR                   1670214 non-null  int64
 2   NAME_CONTRACT_TYPE           1670214 non-null  object
 3   AMT_ANNUITY                  1297979 non-null  float64
 4   AMT_APPLICATION              1670214 non-null  float64
 5   AMT_CREDIT                   1670213 non-null  float64
 6   AMT_DOWN_PAYMENT             774370 non-null   float64
 7   AMT_GOODS_PRICE              1284699 non-null  float64
 8   WEEKDAY_APPR_PROCESS_START   1670214 non-null  object
 9   HOUR_APPR_PROCESS_START      1670214 non-null  int64
 10  FLAG_LAST_APPL_PER_CONTRACT  1670214 non-null  object
 11  NFLAG_LAST_APPL_IN_DAY       1670214 non-null  int64
 12  RATE_DOWN_PAYMENT            774370 non-null   float64
 13  RATE_INTEREST_PRIMARY        5951 non-null     float64
 14  RATE_INTEREST_PRIVILEGED     5951 non-null     float64
 15  NAME_CASH_LOAN_PURPOSE       1670214 non-null  object
 16  NAME_CONTRACT_STATUS         1670214 non-null  object
 17  DAYS_DECISION                1670214 non-null  int64
 18  NAME_PAYMENT_TYPE            1670214 non-null  object
 19  CODE_REJECT_REASON           1670214 non-null  object
 20  NAME_TYPE_SUITE              849809 non-null   object
 21  NAME_CLIENT_TYPE             1670214 non-null  object
 22  NAME_GOODS_CATEGORY          1670214 non-null  object
 23  NAME_PORTFOLIO               1670214 non-null  object
 24  NAME_PRODUCT_TYPE            1670214 non-null  object
 25  CHANNEL_TYPE                 1670214 non-null  object
 26  SELLERPLACE_AREA             1670214 non-null  int64
 27  NAME_SELLER_INDUSTRY         1670214 non-null  object
 28  CNT_PAYMENT                  1297984 non-null  float64
 29  NAME_YIELD_GROUP             1670214 non-null  object
 30  PRODUCT_COMBINATION          1669868 non-null  object
 31  DAYS_FIRST_DRAWING           997149 non-null   float64
 32  DAYS_FIRST_DUE               997149 non-null   float64
 33  DAYS_LAST_DUE_1ST_VERSION    997149 non-null   float64
 34  DAYS_LAST_DUE                997149 non-null   float64
 35  DAYS_TERMINATION             997149 non-null   float64
```

```
 36  NFLAG_INSURED_ON_APPROVAL    997149 non-null   float64
dtypes: float64(15), int64(6), object(16)
memory usage: 471.5+ MB
None

   SK_ID_PREV  SK_ID_CURR NAME_CONTRACT_TYPE  AMT_ANNUITY
AMT_APPLICATION  \
0     2030495      271877     Consumer loans     1730.430
17145.0
1     2802425      108129         Cash loans    25188.615
607500.0
2     2523466      122040         Cash loans    15060.735
112500.0
3     2819243      176158         Cash loans    47041.335
450000.0
4     1784265      202054         Cash loans    31924.395
337500.0

   AMT_CREDIT  AMT_DOWN_PAYMENT  AMT_GOODS_PRICE
WEEKDAY_APPR_PROCESS_START  \
0     17145.0               0.0          17145.0
SATURDAY
1    679671.0               NaN         607500.0
THURSDAY
2    136444.5               NaN         112500.0
TUESDAY
3    470790.0               NaN         450000.0
MONDAY
4    404055.0               NaN         337500.0
THURSDAY

   HOUR_APPR_PROCESS_START  ... NAME_SELLER_INDUSTRY  CNT_PAYMENT  \
0                       15  ...         Connectivity         12.0
1                       11  ...                  XNA         36.0
2                       11  ...                  XNA         12.0
3                        7  ...                  XNA         12.0
4                        9  ...                  XNA         24.0

   NAME_YIELD_GROUP       PRODUCT_COMBINATION  DAYS_FIRST_DRAWING  \
0            middle  POS mobile with interest            365243.0
1       low_action          Cash X-Sell: low            365243.0
2              high         Cash X-Sell: high            365243.0
3            middle       Cash X-Sell: middle            365243.0
4              high         Cash Street: high                 NaN

  DAYS_FIRST_DUE DAYS_LAST_DUE_1ST_VERSION  DAYS_LAST_DUE
DAYS_TERMINATION  \
0          -42.0                     300.0          -42.0            -
37.0
1         -134.0                     916.0       365243.0
```

```
365243.0
2         -271.0                      59.0       365243.0
365243.0
3         -482.0                    -152.0         -182.0           -
177.0
4            NaN                       NaN            NaN
NaN

  NFLAG_INSURED_ON_APPROVAL
0                       0.0
1                       1.0
2                       1.0
3                       1.0
4                       NaN

[5 rows x 37 columns]

         SK_ID_PREV    SK_ID_CURR   AMT_ANNUITY  AMT_APPLICATION  \
count  1.670214e+06  1.670214e+06  1.297979e+06     1.670214e+06
mean   1.923089e+06  2.783572e+05  1.595512e+04     1.752339e+05
std    5.325980e+05  1.028148e+05  1.478214e+04     2.927798e+05
min    1.000001e+06  1.000010e+05  0.000000e+00     0.000000e+00
25%    1.461857e+06  1.893290e+05  6.321780e+03     1.872000e+04
50%    1.923110e+06  2.787145e+05  1.125000e+04     7.104600e+04
75%    2.384280e+06  3.675140e+05  2.065842e+04     1.803600e+05
max    2.845382e+06  4.562550e+05  4.180581e+05     6.905160e+06

         AMT_CREDIT  AMT_DOWN_PAYMENT  AMT_GOODS_PRICE  \
count  1.670213e+06      7.743700e+05     1.284699e+06
mean   1.961140e+05      6.697402e+03     2.278473e+05
std    3.185746e+05      2.092150e+04     3.153966e+05
min    0.000000e+00     -9.000000e-01     0.000000e+00
25%    2.416050e+04      0.000000e+00     5.084100e+04
50%    8.054100e+04      1.638000e+03     1.123200e+05
75%    2.164185e+05      7.740000e+03     2.340000e+05
max    6.905160e+06      3.060045e+06     6.905160e+06

       HOUR_APPR_PROCESS_START  NFLAG_LAST_APPL_IN_DAY
RATE_DOWN_PAYMENT  \
count             1.670214e+06            1.670214e+06
774370.000000
mean              1.248418e+01            9.964675e-01
0.079637
std               3.334028e+00            5.932963e-02
0.107823
min               0.000000e+00            0.000000e+00          -
0.000015
25%               1.000000e+01            1.000000e+00
0.000000
50%               1.200000e+01            1.000000e+00
```

```
0.051605
75%               1.500000e+01            1.000000e+00
0.108909
max               2.300000e+01            1.000000e+00
1.000000

       ...  RATE_INTEREST_PRIVILEGED  DAYS_DECISION  SELLERPLACE_AREA
\
count  ...                5951.000000   1.670214e+06      1.670214e+06

mean   ...                   0.773503  -8.806797e+02      3.139511e+02

std    ...                   0.100879   7.790997e+02      7.127443e+03

min    ...                   0.373150  -2.922000e+03     -1.000000e+00

25%    ...                   0.715645  -1.300000e+03     -1.000000e+00

50%    ...                   0.835095  -5.810000e+02      3.000000e+00

75%    ...                   0.852537  -2.800000e+02      8.200000e+01

max    ...                   1.000000  -1.000000e+00      4.000000e+06


        CNT_PAYMENT  DAYS_FIRST_DRAWING  DAYS_FIRST_DUE  \
count  1.297984e+06       997149.000000   997149.000000
mean   1.605408e+01       342209.855039    13826.269337
std    1.456729e+01        88916.115834    72444.869708
min    0.000000e+00        -2922.000000    -2892.000000
25%    6.000000e+00       365243.000000    -1628.000000
50%    1.200000e+01       365243.000000     -831.000000
75%    2.400000e+01       365243.000000     -411.000000
max    8.400000e+01       365243.000000   365243.000000

       DAYS_LAST_DUE_1ST_VERSION  DAYS_LAST_DUE  DAYS_TERMINATION  \
count                997149.000000  997149.000000     997149.000000
mean                  33767.774054   76582.403064      81992.343838
std                  106857.034789  149647.415123     153303.516729
min                   -2801.000000   -2889.000000      -2874.000000
25%                   -1242.000000   -1314.000000      -1270.000000
50%                    -361.000000    -537.000000       -499.000000
75%                     129.000000     -74.000000        -44.000000
max                  365243.000000  365243.000000     365243.000000

       NFLAG_INSURED_ON_APPROVAL
count                997149.000000
mean                      0.332570
std                       0.471134
```

```
min                     0.000000
25%                     0.000000
50%                     0.000000
75%                     1.000000
max                     1.000000

[8 rows x 21 columns]
```

```
SK_ID_PREV                         0
SK_ID_CURR                         0
NAME_CONTRACT_TYPE                 0
AMT_ANNUITY                   372235
AMT_APPLICATION                    0
AMT_CREDIT                         1
AMT_DOWN_PAYMENT              895844
AMT_GOODS_PRICE               385515
WEEKDAY_APPR_PROCESS_START         0
HOUR_APPR_PROCESS_START            0
FLAG_LAST_APPL_PER_CONTRACT        0
NFLAG_LAST_APPL_IN_DAY             0
RATE_DOWN_PAYMENT             895844
RATE_INTEREST_PRIMARY        1664263
RATE_INTEREST_PRIVILEGED     1664263
NAME_CASH_LOAN_PURPOSE             0
NAME_CONTRACT_STATUS               0
DAYS_DECISION                      0
NAME_PAYMENT_TYPE                  0
CODE_REJECT_REASON                 0
NAME_TYPE_SUITE               820405
NAME_CLIENT_TYPE                   0
NAME_GOODS_CATEGORY                0
NAME_PORTFOLIO                     0
NAME_PRODUCT_TYPE                  0
CHANNEL_TYPE                       0
SELLERPLACE_AREA                   0
NAME_SELLER_INDUSTRY               0
CNT_PAYMENT                   372230
NAME_YIELD_GROUP                   0
PRODUCT_COMBINATION              346
DAYS_FIRST_DRAWING            673065
DAYS_FIRST_DUE                673065
DAYS_LAST_DUE_1ST_VERSION     673065
DAYS_LAST_DUE                 673065
DAYS_TERMINATION              673065
NFLAG_INSURED_ON_APPROVAL     673065
dtype: int64
```

```
POS_CASH_balance: shape is (10001358, 8)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10001358 entries, 0 to 10001357
```

```
Data columns (total 8 columns):
 #   Column                 Dtype
---  ------                 -----
 0   SK_ID_PREV             int64
 1   SK_ID_CURR             int64
 2   MONTHS_BALANCE         int64
 3   CNT_INSTALMENT         float64
 4   CNT_INSTALMENT_FUTURE  float64
 5   NAME_CONTRACT_STATUS   object
 6   SK_DPD                 int64
 7   SK_DPD_DEF             int64
dtypes: float64(2), int64(5), object(1)
memory usage: 610.4+ MB
None
```

	SK_ID_PREV	SK_ID_CURR	MONTHS_BALANCE	CNT_INSTALMENT	\
0	1803195	182943	-31	48.0	
1	1715348	367990	-33	36.0	
2	1784872	397406	-32	12.0	
3	1903291	269225	-35	48.0	
4	2341044	334279	-35	36.0	

	CNT_INSTALMENT_FUTURE	NAME_CONTRACT_STATUS	SK_DPD	SK_DPD_DEF
0	45.0	Active	0	0
1	35.0	Active	0	0
2	9.0	Active	0	0
3	42.0	Active	0	0
4	35.0	Active	0	0

	SK_ID_PREV	SK_ID_CURR	MONTHS_BALANCE	CNT_INSTALMENT	\
count	1.000136e+07	1.000136e+07	1.000136e+07	9.975287e+06	
mean	1.903217e+06	2.784039e+05	-3.501259e+01	1.708965e+01	
std	5.358465e+05	1.027637e+05	2.606657e+01	1.199506e+01	
min	1.000001e+06	1.000010e+05	-9.600000e+01	1.000000e+00	
25%	1.434405e+06	1.895500e+05	-5.400000e+01	1.000000e+01	
50%	1.896565e+06	2.786540e+05	-2.800000e+01	1.200000e+01	
75%	2.368963e+06	3.674290e+05	-1.300000e+01	2.400000e+01	
max	2.843499e+06	4.562550e+05	-1.000000e+00	9.200000e+01	

	CNT_INSTALMENT_FUTURE	SK_DPD	SK_DPD_DEF
count	9.975271e+06	1.000136e+07	1.000136e+07
mean	1.048384e+01	1.160693e+01	6.544684e-01
std	1.110906e+01	1.327140e+02	3.276249e+01
min	0.000000e+00	0.000000e+00	0.000000e+00
25%	3.000000e+00	0.000000e+00	0.000000e+00
50%	7.000000e+00	0.000000e+00	0.000000e+00
75%	1.400000e+01	0.000000e+00	0.000000e+00
max	8.500000e+01	4.231000e+03	3.595000e+03

```
SK_ID_PREV                  0
SK_ID_CURR                  0
```

```
MONTHS_BALANCE              0
CNT_INSTALMENT          26071
CNT_INSTALMENT_FUTURE   26087
NAME_CONTRACT_STATUS        0
SK_DPD                      0
SK_DPD_DEF                  0
dtype: int64

CPU times: user 53.6 s, sys: 5.66 s, total: 59.3 s
Wall time: 1min 15s
```

```
for ds_name in datasets.keys():
    print(f'dataset {ds_name:24}: [ {datasets[ds_name].shape[0]:10,},
{datasets[ds_name].shape[1]}]')
```

```
dataset application_train       : [    307,511, 122]
dataset application_test        : [     48,744, 121]
dataset bureau                  : [  1,716,428, 17]
dataset bureau_balance          : [ 27,299,925, 3]
dataset credit_card_balance     : [  3,840,312, 23]
dataset installments_payments   : [ 13,605,401, 8]
dataset previous_application    : [  1,670,214, 37]
dataset POS_CASH_balance        : [ 10,001,358, 8]
```

One Click Setup | Imports | Load datasets

Below cells are redundant and are added to quickly load all datasets in events like kernel failure.

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import LabelEncoder
import os
import zipfile
from sklearn.base import BaseEstimator, TransformerMixin
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import MinMaxScaler
from sklearn.pipeline import Pipeline, FeatureUnion
from pandas.plotting import scatter_matrix
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import OneHotEncoder
import warnings
warnings.filterwarnings('ignore')
```

```
from google.colab import drive,files
drive.mount('/content/gdrive')

# Google collab dir: Account: kikarand@iu.edu
DATA_DIR="gdrive/MyDrive/data/"


def load_data(in_path, name):
    df = pd.read_csv(in_path)
    print(f"{name}: shape is {df.shape}")
    print(df.info())
    display(df.head(5))
    display(df.describe())
    display(df.isna().sum())
    return df

datasets={}  # lets store the datasets in a dictionary so we can keep
track of them easily

ds_names = ("application_train", "application_test",
"bureau","bureau_balance","credit_card_balance","installments_payments
",
           "previous_application","POS_CASH_balance")

for ds_name in ds_names:
  datasets[ds_name] = load_data(os.path.join(DATA_DIR,
f'{ds_name}.csv'), ds_name)


for ds_name in datasets.keys():
    print(f'dataset {ds_name:24}: [ {datasets[ds_name].shape[0]:10,},
{datasets[ds_name].shape[1]}]')


pa = datasets["previous_application"]
ip = datasets["installments_payments"]
pcb = datasets["POS_CASH_balance"]
ccb = datasets["credit_card_balance"]
bur = datasets["bureau"]
bb = datasets["bureau_balance"]
appsDF = datasets["previous_application"]
```

```
Mounted at /content/gdrive
application_train: shape is (307511, 122)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Columns: 122 entries, SK_ID_CURR to AMT_REQ_CREDIT_BUREAU_YEAR
dtypes: float64(65), int64(41), object(16)
memory usage: 286.2+ MB
None
```

```
   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]

          SK_ID_CURR         TARGET   CNT_CHILDREN
AMT_INCOME_TOTAL  \
count  307511.000000  307511.000000  307511.000000      3.075110e+05

mean   278180.518577       0.080729       0.417052      1.687979e+05

std    102790.175348       0.272419       0.722121      2.371231e+05

min    100002.000000       0.000000       0.000000      2.565000e+04

25%    189145.500000       0.000000       0.000000      1.125000e+05

50%    278202.000000       0.000000       0.000000      1.471500e+05

75%    367142.500000       0.000000       1.000000      2.025000e+05

max    456255.000000       1.000000      19.000000      1.170000e+08


         AMT_CREDIT    AMT_ANNUITY  AMT_GOODS_PRICE  \
count  3.075110e+05  307499.000000     3.072330e+05
mean   5.990260e+05   27108.573909     5.383962e+05
std    4.024908e+05   14493.737315     3.694465e+05
min    4.500000e+04    1615.500000     4.050000e+04
25%    2.700000e+05   16524.000000     2.385000e+05
50%    5.135310e+05   24903.000000     4.500000e+05
75%    8.086500e+05   34596.000000     6.795000e+05
max    4.050000e+06  258025.500000     4.050000e+06

       REGION_POPULATION_RELATIVE     DAYS_BIRTH
DAYS_EMPLOYED  ...  \
count               307511.000000  307511.000000  307511.000000  ...

mean                     0.020868  -16036.995067   63815.045904  ...

std                      0.013831    4363.988632  141275.766519  ...

min                      0.000290  -25229.000000  -17912.000000  ...

25%                      0.010006  -19682.000000   -2760.000000  ...

50%                      0.018850  -15750.000000   -1213.000000  ...
```

```
75%                     0.028663  -12413.000000    -289.000000  ...

max                     0.072508   -7489.000000  365243.000000  ...


       FLAG_DOCUMENT_18  FLAG_DOCUMENT_19  FLAG_DOCUMENT_20
FLAG_DOCUMENT_21  \
count     307511.000000     307511.000000     307511.000000
307511.000000
mean           0.008130          0.000595          0.000507
0.000335
std            0.089798          0.024387          0.022518
0.018299
min            0.000000          0.000000          0.000000
0.000000
25%            0.000000          0.000000          0.000000
0.000000
50%            0.000000          0.000000          0.000000
0.000000
75%            0.000000          0.000000          0.000000
0.000000
max            1.000000          1.000000          1.000000
1.000000

       AMT_REQ_CREDIT_BUREAU_HOUR  AMT_REQ_CREDIT_BUREAU_DAY  \
count                265992.000000              265992.000000
mean                      0.006402                   0.007000
std                       0.083849                   0.110757
min                       0.000000                   0.000000
25%                       0.000000                   0.000000
50%                       0.000000                   0.000000
75%                       0.000000                   0.000000
max                       4.000000                   9.000000

       AMT_REQ_CREDIT_BUREAU_WEEK  AMT_REQ_CREDIT_BUREAU_MON  \
count                265992.000000              265992.000000
mean                      0.034362                   0.267395
std                       0.204685                   0.916002
min                       0.000000                   0.000000
25%                       0.000000                   0.000000
50%                       0.000000                   0.000000
75%                       0.000000                   0.000000
max                       8.000000                  27.000000

       AMT_REQ_CREDIT_BUREAU_QRT  AMT_REQ_CREDIT_BUREAU_YEAR
count                265992.000000              265992.000000
mean                      0.265474                   1.899974
std                       0.794056                   1.869295
min                       0.000000                   0.000000
25%                       0.000000                   0.000000
```

```
50%                      0.000000                     1.000000
75%                      0.000000                     3.000000
max                    261.000000                    25.000000

[8 rows x 106 columns]

SK_ID_CURR                        0
TARGET                            0
NAME_CONTRACT_TYPE                0
CODE_GENDER                       0
FLAG_OWN_CAR                      0
                                ...
AMT_REQ_CREDIT_BUREAU_DAY     41519
AMT_REQ_CREDIT_BUREAU_WEEK    41519
AMT_REQ_CREDIT_BUREAU_MON     41519
AMT_REQ_CREDIT_BUREAU_QRT     41519
AMT_REQ_CREDIT_BUREAU_YEAR    41519
Length: 122, dtype: int64

application_test: shape is (48744, 121)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48744 entries, 0 to 48743
Columns: 121 entries, SK_ID_CURR to AMT_REQ_CREDIT_BUREAU_YEAR
dtypes: float64(65), int64(40), object(16)
memory usage: 45.0+ MB
None

   SK_ID_CURR NAME_CONTRACT_TYPE CODE_GENDER FLAG_OWN_CAR
FLAG_OWN_REALTY  \
0      100001         Cash loans           F            N
Y
1      100005         Cash loans           M            N
Y
2      100013         Cash loans           M            Y
Y
3      100028         Cash loans           F            N
Y
4      100038         Cash loans           M            Y
N

   CNT_CHILDREN  AMT_INCOME_TOTAL  AMT_CREDIT  AMT_ANNUITY
AMT_GOODS_PRICE  \
0             0          135000.0    568800.0      20560.5
450000.0
1             0           99000.0    222768.0      17370.0
180000.0
2             0          202500.0    663264.0      69777.0
630000.0
3             2          315000.0   1575000.0      49018.5
1575000.0
4             1          180000.0    625500.0      32067.0
```

```
625500.0

   ... 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                        0.0                        0.0
4                        NaN                        NaN

   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                         0.0                        0.0
4                         NaN                        NaN

   AMT_REQ_CREDIT_BUREAU_QRT  AMT_REQ_CREDIT_BUREAU_YEAR
0                        0.0                        0.0
1                        0.0                        3.0
2                        1.0                        4.0
3                        0.0                        3.0
4                        NaN                        NaN

[5 rows x 121 columns]

          SK_ID_CURR  CNT_CHILDREN  AMT_INCOME_TOTAL    AMT_CREDIT  \
count   48744.000000  48744.000000      4.874400e+04  4.874400e+04
mean   277796.676350      0.397054      1.784318e+05  5.167404e+05
std    103169.547296      0.709047      1.015226e+05  3.653970e+05
min    100001.000000      0.000000      2.694150e+04  4.500000e+04
25%    188557.750000      0.000000      1.125000e+05  2.606400e+05
50%    277549.000000      0.000000      1.575000e+05  4.500000e+05
75%    367555.500000      1.000000      2.250000e+05  6.750000e+05
max    456250.000000     20.000000      4.410000e+06  2.245500e+06

         AMT_ANNUITY  AMT_GOODS_PRICE  REGION_POPULATION_RELATIVE  \
count   48720.000000     4.874400e+04                48744.000000
```

```
mean    29426.240209     4.626188e+05                    0.021226
std     16013.368315     3.367102e+05                    0.014428
min      2295.000000     4.500000e+04                    0.000253
25%     17973.000000     2.250000e+05                    0.010006
50%     26199.000000     3.960000e+05                    0.018850
75%     37390.500000     6.300000e+05                    0.028663
max    180576.000000     2.245500e+06                    0.072508

         DAYS_BIRTH  DAYS_EMPLOYED  DAYS_REGISTRATION  ...
FLAG_DOCUMENT_18  \
count  48744.000000   48744.000000       48744.000000  ...
48744.000000
mean  -16068.084605   67485.366322       -4967.652716  ...
0.001559
std     4325.900393  144348.507136        3552.612035  ...
0.039456
min   -25195.000000  -17463.000000      -23722.000000  ...
0.000000
25%   -19637.000000   -2910.000000       -7459.250000  ...
0.000000
50%   -15785.000000   -1293.000000       -4490.000000  ...
0.000000
75%   -12496.000000    -296.000000       -1901.000000  ...
0.000000
max    -7338.000000  365243.000000           0.000000  ...
1.000000

       FLAG_DOCUMENT_19  FLAG_DOCUMENT_20  FLAG_DOCUMENT_21  \
count           48744.0           48744.0           48744.0
mean                0.0               0.0               0.0
std                 0.0               0.0               0.0
min                 0.0               0.0               0.0
25%                 0.0               0.0               0.0
50%                 0.0               0.0               0.0
75%                 0.0               0.0               0.0
max                 0.0               0.0               0.0

       AMT_REQ_CREDIT_BUREAU_HOUR  AMT_REQ_CREDIT_BUREAU_DAY  \
count                42695.000000               42695.000000
mean                     0.002108                   0.001803
std                      0.046373                   0.046132
min                      0.000000                   0.000000
25%                      0.000000                   0.000000
50%                      0.000000                   0.000000
75%                      0.000000                   0.000000
max                      2.000000                   2.000000

       AMT_REQ_CREDIT_BUREAU_WEEK  AMT_REQ_CREDIT_BUREAU_MON  \
count                42695.000000               42695.000000
mean                     0.002787                   0.009299
```

```
std                        0.054037                    0.110924
min                        0.000000                    0.000000
25%                        0.000000                    0.000000
50%                        0.000000                    0.000000
75%                        0.000000                    0.000000
max                        2.000000                    6.000000

       AMT_REQ_CREDIT_BUREAU_QRT  AMT_REQ_CREDIT_BUREAU_YEAR
count               42695.000000                42695.000000
mean                    0.546902                    1.983769
std                     0.693305                    1.838873
min                     0.000000                    0.000000
25%                     0.000000                    0.000000
50%                     0.000000                    2.000000
75%                     1.000000                    3.000000
max                     7.000000                   17.000000

[8 rows x 105 columns]

SK_ID_CURR                      0
NAME_CONTRACT_TYPE              0
CODE_GENDER                     0
FLAG_OWN_CAR                    0
FLAG_OWN_REALTY                 0
                             ...
AMT_REQ_CREDIT_BUREAU_DAY    6049
AMT_REQ_CREDIT_BUREAU_WEEK   6049
AMT_REQ_CREDIT_BUREAU_MON    6049
AMT_REQ_CREDIT_BUREAU_QRT    6049
AMT_REQ_CREDIT_BUREAU_YEAR   6049
Length: 121, dtype: int64

bureau: shape is (1716428, 17)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1716428 entries, 0 to 1716427
Data columns (total 17 columns):
 #   Column                  Dtype
---  ------                  -----
 0   SK_ID_CURR              int64
 1   SK_ID_BUREAU            int64
 2   CREDIT_ACTIVE           object
 3   CREDIT_CURRENCY         object
 4   DAYS_CREDIT             int64
 5   CREDIT_DAY_OVERDUE      int64
 6   DAYS_CREDIT_ENDDATE     float64
 7   DAYS_ENDDATE_FACT       float64
 8   AMT_CREDIT_MAX_OVERDUE  float64
 9   CNT_CREDIT_PROLONG      int64
 10  AMT_CREDIT_SUM          float64
 11  AMT_CREDIT_SUM_DEBT     float64
```

```
 12  AMT_CREDIT_SUM_LIMIT    float64
 13  AMT_CREDIT_SUM_OVERDUE  float64
 14  CREDIT_TYPE             object
 15  DAYS_CREDIT_UPDATE      int64
 16  AMT_ANNUITY             float64
dtypes: float64(8), int64(6), object(3)
memory usage: 222.6+ MB
None
```

	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

	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

	AMT_CREDIT_MAX_OVERDUE	CNT_CREDIT_PROLONG	AMT_CREDIT_SUM
0	NaN	0	91323.0
1	NaN	0	225000.0
2	NaN	0	464323.5
3	NaN	0	90000.0
4	77674.5	0	2700000.0

	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

```
       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

         SK_ID_CURR  SK_ID_BUREAU   DAYS_CREDIT  CREDIT_DAY_OVERDUE  \
count  1.716428e+06  1.716428e+06  1.716428e+06        1.716428e+06
mean   2.782149e+05  5.924434e+06 -1.142108e+03        8.181666e-01
std    1.029386e+05  5.322657e+05  7.951649e+02        3.654443e+01
min    1.000010e+05  5.000000e+06 -2.922000e+03        0.000000e+00
25%    1.888668e+05  5.463954e+06 -1.666000e+03        0.000000e+00
50%    2.780550e+05  5.926304e+06 -9.870000e+02        0.000000e+00
75%    3.674260e+05  6.385681e+06 -4.740000e+02        0.000000e+00
max    4.562550e+05  6.843457e+06  0.000000e+00        2.792000e+03

       DAYS_CREDIT_ENDDATE  DAYS_ENDDATE_FACT  AMT_CREDIT_MAX_OVERDUE
\
count         1.610875e+06       1.082775e+06            5.919400e+05

mean          5.105174e+02      -1.017437e+03            3.825418e+03

std           4.994220e+03       7.140106e+02            2.060316e+05

min          -4.206000e+04      -4.202300e+04            0.000000e+00

25%          -1.138000e+03      -1.489000e+03            0.000000e+00

50%          -3.300000e+02      -8.970000e+02            0.000000e+00

75%           4.740000e+02      -4.250000e+02            0.000000e+00

max           3.119900e+04       0.000000e+00            1.159872e+08


       CNT_CREDIT_PROLONG  AMT_CREDIT_SUM  AMT_CREDIT_SUM_DEBT  \
count        1.716428e+06    1.716415e+06         1.458759e+06
mean         6.410406e-03    3.549946e+05         1.370851e+05
std          9.622391e-02    1.149811e+06         6.774011e+05
min          0.000000e+00    0.000000e+00        -4.705600e+06
25%          0.000000e+00    5.130000e+04         0.000000e+00
50%          0.000000e+00    1.255185e+05         0.000000e+00
75%          0.000000e+00    3.150000e+05         4.015350e+04
max          9.000000e+00    5.850000e+08         1.701000e+08

       AMT_CREDIT_SUM_LIMIT  AMT_CREDIT_SUM_OVERDUE
DAYS_CREDIT_UPDATE  \
count          1.124648e+06            1.716428e+06
```

```
1.716428e+06
mean           6.229515e+03            3.791276e+01       -
5.937483e+02
std            4.503203e+04            5.937650e+03
7.207473e+02
min           -5.864061e+05            0.000000e+00       -
4.194700e+04
25%            0.000000e+00            0.000000e+00       -
9.080000e+02
50%            0.000000e+00            0.000000e+00       -
3.950000e+02
75%            0.000000e+00            0.000000e+00       -
3.300000e+01
max            4.705600e+06            3.756681e+06
3.720000e+02

        AMT_ANNUITY
count  4.896370e+05
mean   1.571276e+04
std    3.258269e+05
min    0.000000e+00
25%    0.000000e+00
50%    0.000000e+00
75%    1.350000e+04
max    1.184534e+08

SK_ID_CURR                     0
SK_ID_BUREAU                   0
CREDIT_ACTIVE                  0
CREDIT_CURRENCY                0
DAYS_CREDIT                    0
CREDIT_DAY_OVERDUE             0
DAYS_CREDIT_ENDDATE       105553
DAYS_ENDDATE_FACT         633653
AMT_CREDIT_MAX_OVERDUE   1124488
CNT_CREDIT_PROLONG             0
AMT_CREDIT_SUM                13
AMT_CREDIT_SUM_DEBT       257669
AMT_CREDIT_SUM_LIMIT      591780
AMT_CREDIT_SUM_OVERDUE         0
CREDIT_TYPE                    0
DAYS_CREDIT_UPDATE             0
AMT_ANNUITY              1226791
dtype: int64

bureau_balance: shape is (27299925, 3)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27299925 entries, 0 to 27299924
Data columns (total 3 columns):
 #   Column          Dtype
```

```
---  ------           -----
 0   SK_ID_BUREAU     int64
 1   MONTHS_BALANCE   int64
 2   STATUS           object
dtypes: int64(2), object(1)
memory usage: 624.8+ MB
None
```

	SK_ID_BUREAU	MONTHS_BALANCE	STATUS
0	5715448	0	C
1	5715448	-1	C
2	5715448	-2	C
3	5715448	-3	C
4	5715448	-4	C

	SK_ID_BUREAU	MONTHS_BALANCE
count	2.729992e+07	2.729992e+07
mean	6.036297e+06	-3.074169e+01
std	4.923489e+05	2.386451e+01
min	5.001709e+06	-9.600000e+01
25%	5.730933e+06	-4.600000e+01
50%	6.070821e+06	-2.500000e+01
75%	6.431951e+06	-1.100000e+01
max	6.842888e+06	0.000000e+00

```
SK_ID_BUREAU      0
MONTHS_BALANCE    0
STATUS            0
dtype: int64

credit_card_balance: shape is (3840312, 23)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3840312 entries, 0 to 3840311
Data columns (total 23 columns):
 #   Column                       Dtype
---  ------                       -----
 0   SK_ID_PREV                   int64
 1   SK_ID_CURR                   int64
 2   MONTHS_BALANCE               int64
 3   AMT_BALANCE                  float64
 4   AMT_CREDIT_LIMIT_ACTUAL      int64
 5   AMT_DRAWINGS_ATM_CURRENT     float64
 6   AMT_DRAWINGS_CURRENT         float64
 7   AMT_DRAWINGS_OTHER_CURRENT   float64
 8   AMT_DRAWINGS_POS_CURRENT     float64
 9   AMT_INST_MIN_REGULARITY      float64
 10  AMT_PAYMENT_CURRENT          float64
 11  AMT_PAYMENT_TOTAL_CURRENT    float64
 12  AMT_RECEIVABLE_PRINCIPAL     float64
 13  AMT_RECIVABLE                float64
 14  AMT_TOTAL_RECEIVABLE         float64
```

```
 15  CNT_DRAWINGS_ATM_CURRENT    float64
 16  CNT_DRAWINGS_CURRENT        int64
 17  CNT_DRAWINGS_OTHER_CURRENT  float64
 18  CNT_DRAWINGS_POS_CURRENT    float64
 19  CNT_INSTALMENT_MATURE_CUM   float64
 20  NAME_CONTRACT_STATUS        object
 21  SK_DPD                      int64
 22  SK_DPD_DEF                  int64
dtypes: float64(15), int64(7), object(1)
memory usage: 673.9+ MB
None

   SK_ID_PREV  SK_ID_CURR  MONTHS_BALANCE  AMT_BALANCE  \
0     2562384      378907              -6       56.970
1     2582071      363914              -1    63975.555
2     1740877      371185              -7    31815.225
3     1389973      337855              -4   236572.110
4     1891521      126868              -1   453919.455

   AMT_CREDIT_LIMIT_ACTUAL  AMT_DRAWINGS_ATM_CURRENT
AMT_DRAWINGS_CURRENT  \
0                   135000                      0.0
877.5
1                    45000                   2250.0
2250.0
2                   450000                      0.0
0.0
3                   225000                   2250.0
2250.0
4                   450000                      0.0
11547.0

   AMT_DRAWINGS_OTHER_CURRENT  AMT_DRAWINGS_POS_CURRENT  \
0                         0.0                     877.5
1                         0.0                       0.0
2                         0.0                       0.0
3                         0.0                       0.0
4                         0.0                   11547.0

   AMT_INST_MIN_REGULARITY  ...  AMT_RECIVABLE
AMT_TOTAL_RECEIVABLE  \
0                  1700.325  ...          0.000                0.000

1                  2250.000  ...      64875.555            64875.555

2                  2250.000  ...      31460.085            31460.085

3                 11795.760  ...     233048.970           233048.970

4                 22924.890  ...     453919.455           453919.455
```

```
   CNT_DRAWINGS_ATM_CURRENT  CNT_DRAWINGS_CURRENT
CNT_DRAWINGS_OTHER_CURRENT  \
0                       0.0                     1
0.0
1                       1.0                     1
0.0
2                       0.0                     0
0.0
3                       1.0                     1
0.0
4                       0.0                     1
0.0

   CNT_DRAWINGS_POS_CURRENT  CNT_INSTALMENT_MATURE_CUM
NAME_CONTRACT_STATUS  \
0                       1.0                      35.0
Active
1                       0.0                      69.0
Active
2                       0.0                      30.0
Active
3                       0.0                      10.0
Active
4                       1.0                     101.0
Active

   SK_DPD  SK_DPD_DEF
0       0           0
1       0           0
2       0           0
3       0           0
4       0           0

[5 rows x 23 columns]

         SK_ID_PREV    SK_ID_CURR  MONTHS_BALANCE   AMT_BALANCE  \
count  3.840312e+06  3.840312e+06    3.840312e+06  3.840312e+06
mean   1.904504e+06  2.783242e+05   -3.452192e+01  5.830016e+04
std    5.364695e+05  1.027045e+05    2.666775e+01  1.063070e+05
min    1.000018e+06  1.000060e+05   -9.600000e+01 -4.202502e+05
25%    1.434385e+06  1.895170e+05   -5.500000e+01  0.000000e+00
50%    1.897122e+06  2.783960e+05   -2.800000e+01  0.000000e+00
75%    2.369328e+06  3.675800e+05   -1.100000e+01  8.904669e+04
max    2.843496e+06  4.562500e+05   -1.000000e+00  1.505902e+06

       AMT_CREDIT_LIMIT_ACTUAL  AMT_DRAWINGS_ATM_CURRENT  \
count             3.840312e+06              3.090496e+06
mean              1.538080e+05              5.961325e+03
```

```
std               1.651457e+05              2.822569e+04
min               0.000000e+00             -6.827310e+03
25%               4.500000e+04              0.000000e+00
50%               1.125000e+05              0.000000e+00
75%               1.800000e+05              0.000000e+00
max               1.350000e+06              2.115000e+06

       AMT_DRAWINGS_CURRENT  AMT_DRAWINGS_OTHER_CURRENT  \
count          3.840312e+06                3.090496e+06
mean           7.433388e+03                2.881696e+02
std            3.384608e+04                8.201989e+03
min           -6.211620e+03                0.000000e+00
25%            0.000000e+00                0.000000e+00
50%            0.000000e+00                0.000000e+00
75%            0.000000e+00                0.000000e+00
max            2.287098e+06                1.529847e+06

       AMT_DRAWINGS_POS_CURRENT  AMT_INST_MIN_REGULARITY  ...  \
count              3.090496e+06             3.535076e+06  ...
mean               2.968805e+03             3.540204e+03  ...
std                2.079689e+04             5.600154e+03  ...
min                0.000000e+00             0.000000e+00  ...
25%                0.000000e+00             0.000000e+00  ...
50%                0.000000e+00             0.000000e+00  ...
75%                0.000000e+00             6.633911e+03  ...
max                2.239274e+06             2.028820e+05  ...

       AMT_RECEIVABLE_PRINCIPAL  AMT_RECIVABLE
AMT_TOTAL_RECEIVABLE  \
count              3.840312e+06   3.840312e+06          3.840312e+06

mean               5.596588e+04   5.808881e+04          5.809829e+04

std                1.025336e+05   1.059654e+05          1.059718e+05

min               -4.233058e+05  -4.202502e+05         -4.202502e+05

25%                0.000000e+00   0.000000e+00          0.000000e+00

50%                0.000000e+00   0.000000e+00          0.000000e+00

75%                8.535924e+04   8.889949e+04          8.891451e+04

max                1.472317e+06   1.493338e+06          1.493338e+06


       CNT_DRAWINGS_ATM_CURRENT  CNT_DRAWINGS_CURRENT  \
count              3.090496e+06          3.840312e+06
mean               3.094490e-01          7.031439e-01
```

```
std                 1.100401e+00          3.190347e+00
min                 0.000000e+00          0.000000e+00
25%                 0.000000e+00          0.000000e+00
50%                 0.000000e+00          0.000000e+00
75%                 0.000000e+00          0.000000e+00
max                 5.100000e+01          1.650000e+02

       CNT_DRAWINGS_OTHER_CURRENT  CNT_DRAWINGS_POS_CURRENT  \
count                3.090496e+06              3.090496e+06
mean                 4.812496e-03              5.594791e-01
std                  8.263861e-02              3.240649e+00
min                  0.000000e+00              0.000000e+00
25%                  0.000000e+00              0.000000e+00
50%                  0.000000e+00              0.000000e+00
75%                  0.000000e+00              0.000000e+00
max                  1.200000e+01              1.650000e+02

       CNT_INSTALMENT_MATURE_CUM        SK_DPD    SK_DPD_DEF
count               3.535076e+06  3.840312e+06  3.840312e+06
mean                2.082508e+01  9.283667e+00  3.316220e-01
std                 2.005149e+01  9.751570e+01  2.147923e+01
min                 0.000000e+00  0.000000e+00  0.000000e+00
25%                 4.000000e+00  0.000000e+00  0.000000e+00
50%                 1.500000e+01  0.000000e+00  0.000000e+00
75%                 3.200000e+01  0.000000e+00  0.000000e+00
max                 1.200000e+02  3.260000e+03  3.260000e+03

[8 rows x 22 columns]

SK_ID_PREV                         0
SK_ID_CURR                         0
MONTHS_BALANCE                     0
AMT_BALANCE                        0
AMT_CREDIT_LIMIT_ACTUAL            0
AMT_DRAWINGS_ATM_CURRENT      749816
AMT_DRAWINGS_CURRENT               0
AMT_DRAWINGS_OTHER_CURRENT    749816
AMT_DRAWINGS_POS_CURRENT      749816
AMT_INST_MIN_REGULARITY       305236
AMT_PAYMENT_CURRENT           767988
AMT_PAYMENT_TOTAL_CURRENT          0
AMT_RECEIVABLE_PRINCIPAL           0
AMT_RECIVABLE                      0
AMT_TOTAL_RECEIVABLE               0
CNT_DRAWINGS_ATM_CURRENT      749816
CNT_DRAWINGS_CURRENT               0
CNT_DRAWINGS_OTHER_CURRENT    749816
CNT_DRAWINGS_POS_CURRENT      749816
CNT_INSTALMENT_MATURE_CUM     305236
NAME_CONTRACT_STATUS               0
```

```
SK_DPD                          0
SK_DPD_DEF                      0
dtype: int64

installments_payments: shape is (13605401, 8)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13605401 entries, 0 to 13605400
Data columns (total 8 columns):
 #   Column                  Dtype
---  ------                  -----
 0   SK_ID_PREV              int64
 1   SK_ID_CURR              int64
 2   NUM_INSTALMENT_VERSION  float64
 3   NUM_INSTALMENT_NUMBER   int64
 4   DAYS_INSTALMENT         float64
 5   DAYS_ENTRY_PAYMENT      float64
 6   AMT_INSTALMENT          float64
 7   AMT_PAYMENT             float64
dtypes: float64(5), int64(3)
memory usage: 830.4 MB
None

   SK_ID_PREV  SK_ID_CURR  NUM_INSTALMENT_VERSION
NUM_INSTALMENT_NUMBER  \
0     1054186      161674                    1.0
6
1     1330831      151639                    0.0
34
2     2085231      193053                    2.0
1
3     2452527      199697                    1.0
3
4     2714724      167756                    1.0
2

   DAYS_INSTALMENT  DAYS_ENTRY_PAYMENT  AMT_INSTALMENT  AMT_PAYMENT
0          -1180.0             -1187.0        6948.360     6948.360
1          -2156.0             -2156.0        1716.525     1716.525
2            -63.0               -63.0       25425.000    25425.000
3          -2418.0             -2426.0       24350.130    24350.130
4          -1383.0             -1366.0        2165.040     2160.585

         SK_ID_PREV    SK_ID_CURR  NUM_INSTALMENT_VERSION  \
count  1.360540e+07  1.360540e+07            1.360540e+07
mean   1.903365e+06  2.784449e+05            8.566373e-01
std    5.362029e+05  1.027183e+05            1.035216e+00
min    1.000001e+06  1.000010e+05            0.000000e+00
25%    1.434191e+06  1.896390e+05            0.000000e+00
50%    1.896520e+06  2.786850e+05            1.000000e+00
75%    2.369094e+06  3.675300e+05            1.000000e+00
max    2.843499e+06  4.562550e+05            1.780000e+02
```

```
       NUM_INSTALMENT_NUMBER  DAYS_INSTALMENT  DAYS_ENTRY_PAYMENT  \
count           1.360540e+07     1.360540e+07        1.360250e+07
mean            1.887090e+01    -1.042270e+03       -1.051114e+03
std             2.666407e+01     8.009463e+02        8.005859e+02
min             1.000000e+00    -2.922000e+03       -4.921000e+03
25%             4.000000e+00    -1.654000e+03       -1.662000e+03
50%             8.000000e+00    -8.180000e+02       -8.270000e+02
75%             1.900000e+01    -3.610000e+02       -3.700000e+02
max             2.770000e+02    -1.000000e+00       -1.000000e+00

       AMT_INSTALMENT   AMT_PAYMENT
count    1.360540e+07  1.360250e+07
mean     1.705091e+04  1.723822e+04
std      5.057025e+04  5.473578e+04
min      0.000000e+00  0.000000e+00
25%      4.226085e+03  3.398265e+03
50%      8.884080e+03  8.125515e+03
75%      1.671021e+04  1.610842e+04
max      3.771488e+06  3.771488e+06

SK_ID_PREV                   0
SK_ID_CURR                   0
NUM_INSTALMENT_VERSION       0
NUM_INSTALMENT_NUMBER        0
DAYS_INSTALMENT              0
DAYS_ENTRY_PAYMENT        2905
AMT_INSTALMENT               0
AMT_PAYMENT               2905
dtype: int64

previous_application: shape is (1670214, 37)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1670214 entries, 0 to 1670213
Data columns (total 37 columns):
 #   Column                       Non-Null Count    Dtype
---  ------                       --------------    -----
 0   SK_ID_PREV                   1670214 non-null  int64
 1   SK_ID_CURR                   1670214 non-null  int64
 2   NAME_CONTRACT_TYPE           1670214 non-null  object
 3   AMT_ANNUITY                  1297979 non-null  float64
 4   AMT_APPLICATION              1670214 non-null  float64
 5   AMT_CREDIT                   1670213 non-null  float64
 6   AMT_DOWN_PAYMENT             774370 non-null   float64
 7   AMT_GOODS_PRICE              1284699 non-null  float64
 8   WEEKDAY_APPR_PROCESS_START   1670214 non-null  object
 9   HOUR_APPR_PROCESS_START      1670214 non-null  int64
 10  FLAG_LAST_APPL_PER_CONTRACT  1670214 non-null  object
 11  NFLAG_LAST_APPL_IN_DAY       1670214 non-null  int64
 12  RATE_DOWN_PAYMENT            774370 non-null   float64
```

```
 13  RATE_INTEREST_PRIMARY        5951 non-null     float64
 14  RATE_INTEREST_PRIVILEGED     5951 non-null     float64
 15  NAME_CASH_LOAN_PURPOSE       1670214 non-null  object
 16  NAME_CONTRACT_STATUS         1670214 non-null  object
 17  DAYS_DECISION                1670214 non-null  int64
 18  NAME_PAYMENT_TYPE            1670214 non-null  object
 19  CODE_REJECT_REASON           1670214 non-null  object
 20  NAME_TYPE_SUITE              849809 non-null   object
 21  NAME_CLIENT_TYPE             1670214 non-null  object
 22  NAME_GOODS_CATEGORY          1670214 non-null  object
 23  NAME_PORTFOLIO               1670214 non-null  object
 24  NAME_PRODUCT_TYPE            1670214 non-null  object
 25  CHANNEL_TYPE                 1670214 non-null  object
 26  SELLERPLACE_AREA             1670214 non-null  int64
 27  NAME_SELLER_INDUSTRY         1670214 non-null  object
 28  CNT_PAYMENT                  1297984 non-null  float64
 29  NAME_YIELD_GROUP             1670214 non-null  object
 30  PRODUCT_COMBINATION          1669868 non-null  object
 31  DAYS_FIRST_DRAWING           997149 non-null   float64
 32  DAYS_FIRST_DUE               997149 non-null   float64
 33  DAYS_LAST_DUE_1ST_VERSION    997149 non-null   float64
 34  DAYS_LAST_DUE                997149 non-null   float64
 35  DAYS_TERMINATION             997149 non-null   float64
 36  NFLAG_INSURED_ON_APPROVAL    997149 non-null   float64
dtypes: float64(15), int64(6), object(16)
memory usage: 471.5+ MB
None

   SK_ID_PREV  SK_ID_CURR NAME_CONTRACT_TYPE  AMT_ANNUITY
AMT_APPLICATION  \
0     2030495      271877     Consumer loans     1730.430
17145.0
1     2802425      108129         Cash loans    25188.615
607500.0
2     2523466      122040         Cash loans    15060.735
112500.0
3     2819243      176158         Cash loans    47041.335
450000.0
4     1784265      202054         Cash loans    31924.395
337500.0

   AMT_CREDIT  AMT_DOWN_PAYMENT  AMT_GOODS_PRICE
WEEKDAY_APPR_PROCESS_START  \
0     17145.0               0.0          17145.0
SATURDAY
1    679671.0               NaN         607500.0
THURSDAY
2    136444.5               NaN         112500.0
TUESDAY
3    470790.0               NaN         450000.0
```

```
MONDAY
4    404055.0               NaN         337500.0
THURSDAY

   HOUR_APPR_PROCESS_START  ... NAME_SELLER_INDUSTRY  CNT_PAYMENT  \
0                       15  ...         Connectivity         12.0
1                       11  ...                  XNA         36.0
2                       11  ...                  XNA         12.0
3                        7  ...                  XNA         12.0
4                        9  ...                  XNA         24.0

   NAME_YIELD_GROUP       PRODUCT_COMBINATION  DAYS_FIRST_DRAWING  \
0            middle  POS mobile with interest            365243.0
1        low_action          Cash X-Sell: low            365243.0
2              high         Cash X-Sell: high            365243.0
3            middle       Cash X-Sell: middle            365243.0
4              high         Cash Street: high                 NaN

  DAYS_FIRST_DUE DAYS_LAST_DUE_1ST_VERSION  DAYS_LAST_DUE
DAYS_TERMINATION  \
0          -42.0                     300.0          -42.0            -
37.0
1         -134.0                     916.0       365243.0
365243.0
2         -271.0                      59.0       365243.0
365243.0
3         -482.0                    -152.0         -182.0           -
177.0
4            NaN                       NaN            NaN
NaN

  NFLAG_INSURED_ON_APPROVAL
0                       0.0
1                       1.0
2                       1.0
3                       1.0
4                       NaN

[5 rows x 37 columns]

         SK_ID_PREV    SK_ID_CURR   AMT_ANNUITY  AMT_APPLICATION  \
count  1.670214e+06  1.670214e+06  1.297979e+06     1.670214e+06
mean   1.923089e+06  2.783572e+05  1.595512e+04     1.752339e+05
std    5.325980e+05  1.028148e+05  1.478214e+04     2.927798e+05
min    1.000001e+06  1.000010e+05  0.000000e+00     0.000000e+00
25%    1.461857e+06  1.893290e+05  6.321780e+03     1.872000e+04
50%    1.923110e+06  2.787145e+05  1.125000e+04     7.104600e+04
75%    2.384280e+06  3.675140e+05  2.065842e+04     1.803600e+05
max    2.845382e+06  4.562550e+05  4.180581e+05     6.905160e+06
```

```
         AMT_CREDIT  AMT_DOWN_PAYMENT  AMT_GOODS_PRICE  \
count  1.670213e+06      7.743700e+05     1.284699e+06
mean   1.961140e+05      6.697402e+03     2.278473e+05
std    3.185746e+05      2.092150e+04     3.153966e+05
min    0.000000e+00     -9.000000e-01     0.000000e+00
25%    2.416050e+04      0.000000e+00     5.084100e+04
50%    8.054100e+04      1.638000e+03     1.123200e+05
75%    2.164185e+05      7.740000e+03     2.340000e+05
max    6.905160e+06      3.060045e+06     6.905160e+06

       HOUR_APPR_PROCESS_START  NFLAG_LAST_APPL_IN_DAY
RATE_DOWN_PAYMENT  \
count             1.670214e+06            1.670214e+06
774370.000000
mean              1.248418e+01            9.964675e-01
0.079637
std               3.334028e+00            5.932963e-02
0.107823
min               0.000000e+00            0.000000e+00          -
0.000015
25%               1.000000e+01            1.000000e+00
0.000000
50%               1.200000e+01            1.000000e+00
0.051605
75%               1.500000e+01            1.000000e+00
0.108909
max               2.300000e+01            1.000000e+00
1.000000

       ...  RATE_INTEREST_PRIVILEGED  DAYS_DECISION  SELLERPLACE_AREA
\
count  ...               5951.000000   1.670214e+06      1.670214e+06

mean   ...                  0.773503  -8.806797e+02      3.139511e+02

std    ...                  0.100879   7.790997e+02      7.127443e+03

min    ...                  0.373150  -2.922000e+03     -1.000000e+00

25%    ...                  0.715645  -1.300000e+03     -1.000000e+00

50%    ...                  0.835095  -5.810000e+02      3.000000e+00

75%    ...                  0.852537  -2.800000e+02      8.200000e+01

max    ...                  1.000000  -1.000000e+00      4.000000e+06


        CNT_PAYMENT  DAYS_FIRST_DRAWING  DAYS_FIRST_DUE  \
```

```
count  1.297984e+06       997149.000000   997149.000000
mean   1.605408e+01       342209.855039    13826.269337
std    1.456729e+01        88916.115834    72444.869708
min    0.000000e+00        -2922.000000    -2892.000000
25%    6.000000e+00       365243.000000    -1628.000000
50%    1.200000e+01       365243.000000     -831.000000
75%    2.400000e+01       365243.000000     -411.000000
max    8.400000e+01       365243.000000   365243.000000

       DAYS_LAST_DUE_1ST_VERSION  DAYS_LAST_DUE  DAYS_TERMINATION  \
count              997149.000000  997149.000000     997149.000000
mean                33767.774054   76582.403064      81992.343838
std                106857.034789  149647.415123     153303.516729
min                 -2801.000000   -2889.000000      -2874.000000
25%                 -1242.000000   -1314.000000      -1270.000000
50%                  -361.000000    -537.000000       -499.000000
75%                   129.000000     -74.000000        -44.000000
max                365243.000000  365243.000000     365243.000000

       NFLAG_INSURED_ON_APPROVAL
count              997149.000000
mean                    0.332570
std                     0.471134
min                     0.000000
25%                     0.000000
50%                     0.000000
75%                     1.000000
max                     1.000000

[8 rows x 21 columns]
```

```
SK_ID_PREV                          0
SK_ID_CURR                          0
NAME_CONTRACT_TYPE                  0
AMT_ANNUITY                    372235
AMT_APPLICATION                     0
AMT_CREDIT                          1
AMT_DOWN_PAYMENT               895844
AMT_GOODS_PRICE                385515
WEEKDAY_APPR_PROCESS_START          0
HOUR_APPR_PROCESS_START             0
FLAG_LAST_APPL_PER_CONTRACT         0
NFLAG_LAST_APPL_IN_DAY              0
RATE_DOWN_PAYMENT              895844
RATE_INTEREST_PRIMARY         1664263
RATE_INTEREST_PRIVILEGED      1664263
NAME_CASH_LOAN_PURPOSE              0
NAME_CONTRACT_STATUS                0
DAYS_DECISION                       0
NAME_PAYMENT_TYPE                   0
```

```
CODE_REJECT_REASON                  0
NAME_TYPE_SUITE                820405
NAME_CLIENT_TYPE                    0
NAME_GOODS_CATEGORY                 0
NAME_PORTFOLIO                      0
NAME_PRODUCT_TYPE                   0
CHANNEL_TYPE                        0
SELLERPLACE_AREA                    0
NAME_SELLER_INDUSTRY                0
CNT_PAYMENT                    372230
NAME_YIELD_GROUP                    0
PRODUCT_COMBINATION               346
DAYS_FIRST_DRAWING             673065
DAYS_FIRST_DUE                 673065
DAYS_LAST_DUE_1ST_VERSION      673065
DAYS_LAST_DUE                  673065
DAYS_TERMINATION               673065
NFLAG_INSURED_ON_APPROVAL      673065
dtype: int64

POS_CASH_balance: shape is (10001358, 8)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10001358 entries, 0 to 10001357
Data columns (total 8 columns):
 #   Column                 Dtype
---  ------                 -----
 0   SK_ID_PREV             int64
 1   SK_ID_CURR             int64
 2   MONTHS_BALANCE         int64
 3   CNT_INSTALMENT         float64
 4   CNT_INSTALMENT_FUTURE  float64
 5   NAME_CONTRACT_STATUS   object
 6   SK_DPD                 int64
 7   SK_DPD_DEF             int64
dtypes: float64(2), int64(5), object(1)
memory usage: 610.4+ MB
None

   SK_ID_PREV  SK_ID_CURR  MONTHS_BALANCE  CNT_INSTALMENT  \
0     1803195      182943             -31            48.0
1     1715348      367990             -33            36.0
2     1784872      397406             -32            12.0
3     1903291      269225             -35            48.0
4     2341044      334279             -35            36.0

   CNT_INSTALMENT_FUTURE NAME_CONTRACT_STATUS  SK_DPD  SK_DPD_DEF
0                   45.0              Active       0           0
1                   35.0              Active       0           0
2                    9.0              Active       0           0
```

```
3                  42.0               Active       0           0
4                  35.0               Active       0           0

         SK_ID_PREV    SK_ID_CURR  MONTHS_BALANCE  CNT_INSTALMENT  \
count  1.000136e+07  1.000136e+07    1.000136e+07    9.975287e+06
mean   1.903217e+06  2.784039e+05   -3.501259e+01    1.708965e+01
std    5.358465e+05  1.027637e+05    2.606657e+01    1.199506e+01
min    1.000001e+06  1.000010e+05   -9.600000e+01    1.000000e+00
25%    1.434405e+06  1.895500e+05   -5.400000e+01    1.000000e+01
50%    1.896565e+06  2.786540e+05   -2.800000e+01    1.200000e+01
75%    2.368963e+06  3.674290e+05   -1.300000e+01    2.400000e+01
max    2.843499e+06  4.562550e+05   -1.000000e+00    9.200000e+01

       CNT_INSTALMENT_FUTURE        SK_DPD    SK_DPD_DEF
count           9.975271e+06  1.000136e+07  1.000136e+07
mean            1.048384e+01  1.160693e+01  6.544684e-01
std             1.110906e+01  1.327140e+02  3.276249e+01
min             0.000000e+00  0.000000e+00  0.000000e+00
25%             3.000000e+00  0.000000e+00  0.000000e+00
50%             7.000000e+00  0.000000e+00  0.000000e+00
75%             1.400000e+01  0.000000e+00  0.000000e+00
max             8.500000e+01  4.231000e+03  3.595000e+03

SK_ID_PREV                   0
SK_ID_CURR                   0
MONTHS_BALANCE               0
CNT_INSTALMENT           26071
CNT_INSTALMENT_FUTURE    26087
NAME_CONTRACT_STATUS         0
SK_DPD                       0
SK_DPD_DEF                   0
dtype: int64

dataset application_train       : [    307,511, 122]
dataset application_test        : [     48,744, 121]
dataset bureau                  : [  1,716,428, 17]
dataset bureau_balance          : [ 27,299,925, 3]
dataset credit_card_balance     : [  3,840,312, 23]
dataset installments_payments   : [ 13,605,401, 8]
dataset previous_application    : [  1,670,214, 37]
dataset POS_CASH_balance        : [ 10,001,358, 8]
```

Exploratory Data Analysis

Summary of Application train

```
datasets["application_train"].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
```

```
Columns: 122 entries, SK_ID_CURR to AMT_REQ_CREDIT_BUREAU_YEAR
dtypes: float64(65), int64(41), object(16)
memory usage: 286.2+ MB
```

```
datasets["application_train"].describe() #numerical only features
```

```
          SK_ID_CURR         TARGET   CNT_CHILDREN                    
AMT_INCOME_TOTAL  \
count  307511.000000  307511.000000  307511.000000      3.075110e+05  

mean   278180.518577       0.080729       0.417052      1.687979e+05  

std    102790.175348       0.272419       0.722121      2.371231e+05  

min    100002.000000       0.000000       0.000000      2.565000e+04  

25%    189145.500000       0.000000       0.000000      1.125000e+05  

50%    278202.000000       0.000000       0.000000      1.471500e+05  

75%    367142.500000       0.000000       1.000000      2.025000e+05  

max    456255.000000       1.000000      19.000000      1.170000e+08  


         AMT_CREDIT    AMT_ANNUITY  AMT_GOODS_PRICE  \
count  3.075110e+05  307499.000000     3.072330e+05  
mean   5.990260e+05   27108.573909     5.383962e+05  
std    4.024908e+05   14493.737315     3.694465e+05  
min    4.500000e+04    1615.500000     4.050000e+04  
25%    2.700000e+05   16524.000000     2.385000e+05  
50%    5.135310e+05   24903.000000     4.500000e+05  
75%    8.086500e+05   34596.000000     6.795000e+05  
max    4.050000e+06  258025.500000     4.050000e+06  

       REGION_POPULATION_RELATIVE     DAYS_BIRTH                    
DAYS_EMPLOYED  ...  \
count               307511.000000  307511.000000  307511.000000  ...  

mean                     0.020868  -16036.995067   63815.045904  ...  

std                      0.013831    4363.988632  141275.766519  ...  

min                      0.000290  -25229.000000  -17912.000000  ...  

25%                      0.010006  -19682.000000   -2760.000000  ...  

50%                      0.018850  -15750.000000   -1213.000000  ...
```

```
75%                     0.028663  -12413.000000    -289.000000  ...

max                     0.072508   -7489.000000  365243.000000  ...


       FLAG_DOCUMENT_18  FLAG_DOCUMENT_19  FLAG_DOCUMENT_20
FLAG_DOCUMENT_21  \
count     307511.000000     307511.000000     307511.000000
307511.000000
mean           0.008130          0.000595          0.000507
0.000335
std            0.089798          0.024387          0.022518
0.018299
min            0.000000          0.000000          0.000000
0.000000
25%            0.000000          0.000000          0.000000
0.000000
50%            0.000000          0.000000          0.000000
0.000000
75%            0.000000          0.000000          0.000000
0.000000
max            1.000000          1.000000          1.000000
1.000000

       AMT_REQ_CREDIT_BUREAU_HOUR  AMT_REQ_CREDIT_BUREAU_DAY  \
count                265992.000000              265992.000000
mean                      0.006402                   0.007000
std                       0.083849                   0.110757
min                       0.000000                   0.000000
25%                       0.000000                   0.000000
50%                       0.000000                   0.000000
75%                       0.000000                   0.000000
max                       4.000000                   9.000000

       AMT_REQ_CREDIT_BUREAU_WEEK  AMT_REQ_CREDIT_BUREAU_MON  \
count                265992.000000              265992.000000
mean                      0.034362                   0.267395
std                       0.204685                   0.916002
min                       0.000000                   0.000000
25%                       0.000000                   0.000000
50%                       0.000000                   0.000000
75%                       0.000000                   0.000000
max                       8.000000                  27.000000

       AMT_REQ_CREDIT_BUREAU_QRT  AMT_REQ_CREDIT_BUREAU_YEAR
count                265992.000000              265992.000000
mean                      0.265474                   1.899974
std                       0.794056                   1.869295
min                       0.000000                   0.000000
25%                       0.000000                   0.000000
```

```
50%                     0.000000                    1.000000
75%                     0.000000                    3.000000
max                   261.000000                   25.000000

[8 rows x 106 columns]
```

```
datasets["application_test"].describe() #numerical only features
```

```
          SK_ID_CURR  CNT_CHILDREN  AMT_INCOME_TOTAL    AMT_CREDIT  \
count   48744.000000  48744.000000      4.874400e+04  4.874400e+04
mean   277796.676350      0.397054      1.784318e+05  5.167404e+05
std    103169.547296      0.709047      1.015226e+05  3.653970e+05
min    100001.000000      0.000000      2.694150e+04  4.500000e+04
25%    188557.750000      0.000000      1.125000e+05  2.606400e+05
50%    277549.000000      0.000000      1.575000e+05  4.500000e+05
75%    367555.500000      1.000000      2.250000e+05  6.750000e+05
max    456250.000000     20.000000      4.410000e+06  2.245500e+06

          AMT_ANNUITY  AMT_GOODS_PRICE  REGION_POPULATION_RELATIVE  \
count    48720.000000     4.874400e+04                48744.000000
mean     29426.240209     4.626188e+05                    0.021226
std      16016.368315     3.367102e+05                    0.014428
min       2295.000000     4.500000e+04                    0.000253
25%      17973.000000     2.250000e+05                    0.010006
50%      26199.000000     3.960000e+05                    0.018850
75%      37390.500000     6.300000e+05                    0.028663
max     180576.000000     2.245500e+06                    0.072508

          DAYS_BIRTH  DAYS_EMPLOYED  DAYS_REGISTRATION  ...
FLAG_DOCUMENT_18  \
count  48744.000000   48744.000000       48744.000000  ...
48744.000000
mean  -16068.084605   67485.366322       -4967.652716  ...
0.001559
std     4325.900393  144348.507136        3552.612035  ...
0.039456
min   -25195.000000  -17463.000000      -23722.000000  ...
0.000000
25%   -19637.000000   -2910.000000       -7459.250000  ...
0.000000
50%   -15785.000000   -1293.000000       -4490.000000  ...
0.000000
75%   -12496.000000    -296.000000       -1901.000000  ...
0.000000
max    -7338.000000  365243.000000           0.000000  ...
1.000000

       FLAG_DOCUMENT_19  FLAG_DOCUMENT_20  FLAG_DOCUMENT_21  \
count           48744.0           48744.0           48744.0
mean                0.0               0.0               0.0
```

```
std                 0.0               0.0               0.0
min                 0.0               0.0               0.0
25%                 0.0               0.0               0.0
50%                 0.0               0.0               0.0
75%                 0.0               0.0               0.0
max                 0.0               0.0               0.0

       AMT_REQ_CREDIT_BUREAU_HOUR  AMT_REQ_CREDIT_BUREAU_DAY  \
count                42695.000000               42695.000000
mean                     0.002108                   0.001803
std                      0.046373                   0.046132
min                      0.000000                   0.000000
25%                      0.000000                   0.000000
50%                      0.000000                   0.000000
75%                      0.000000                   0.000000
max                      2.000000                   2.000000

       AMT_REQ_CREDIT_BUREAU_WEEK  AMT_REQ_CREDIT_BUREAU_MON  \
count                42695.000000               42695.000000
mean                     0.002787                   0.009299
std                      0.054037                   0.110924
min                      0.000000                   0.000000
25%                      0.000000                   0.000000
50%                      0.000000                   0.000000
75%                      0.000000                   0.000000
max                      2.000000                   6.000000

       AMT_REQ_CREDIT_BUREAU_QRT  AMT_REQ_CREDIT_BUREAU_YEAR
count                42695.000000               42695.000000
mean                     0.546902                   1.983769
std                      0.693305                   1.838873
min                      0.000000                   0.000000
25%                      0.000000                   0.000000
50%                      0.000000                   2.000000
75%                      1.000000                   3.000000
max                      7.000000                  17.000000

[8 rows x 105 columns]
```

```
datasets["application_train"].describe(include='all') #look at all
categorical and numerical
```

```
           SK_ID_CURR         TARGET NAME_CONTRACT_TYPE CODE_GENDER  \
count   307511.000000  307511.000000             307511      307511
unique            NaN            NaN                  2           3
top               NaN            NaN         Cash loans           F
freq              NaN            NaN             278232      202448
mean    278180.518577       0.080729                NaN         NaN
std     102790.175348       0.272419                NaN         NaN
min     100002.000000       0.000000                NaN         NaN
```

```
25%     189145.500000       0.000000                NaN          NaN
50%     278202.000000       0.000000                NaN          NaN
75%     367142.500000       0.000000                NaN          NaN
max     456255.000000       1.000000                NaN          NaN

       FLAG_OWN_CAR FLAG_OWN_REALTY   CNT_CHILDREN
AMT_INCOME_TOTAL  \
count        307511          307511  307511.000000      3.075110e+05

unique            2               2            NaN               NaN

top               N               Y            NaN               NaN

freq         202924          213312            NaN               NaN

mean            NaN             NaN       0.417052      1.687979e+05

std             NaN             NaN       0.722121      2.371231e+05

min             NaN             NaN       0.000000      2.565000e+04

25%             NaN             NaN       0.000000      1.125000e+05

50%             NaN             NaN       0.000000      1.471500e+05

75%             NaN             NaN       1.000000      2.025000e+05

max             NaN             NaN      19.000000      1.170000e+08


          AMT_CREDIT    AMT_ANNUITY  ...  FLAG_DOCUMENT_18
FLAG_DOCUMENT_19  \
count   3.075110e+05  307499.000000  ...     307511.000000
307511.000000
unique           NaN            NaN  ...               NaN
NaN
top              NaN            NaN  ...               NaN
NaN
freq             NaN            NaN  ...               NaN
NaN
mean    5.990260e+05   27108.573909  ...          0.008130
0.000595
std     4.024908e+05   14493.737315  ...          0.089798
0.024387
min     4.500000e+04    1615.500000  ...          0.000000
0.000000
25%     2.700000e+05   16524.000000  ...          0.000000
0.000000
50%     5.135310e+05   24903.000000  ...          0.000000
```

```
0.000000
75%     8.086500e+05   34596.000000  ...          0.000000
0.000000
max     4.050000e+06  258025.500000  ...          1.000000
1.000000

       FLAG_DOCUMENT_20 FLAG_DOCUMENT_21 AMT_REQ_CREDIT_BUREAU_HOUR  \
count     307511.000000    307511.000000               265992.000000
unique              NaN              NaN                         NaN
top                 NaN              NaN                         NaN
freq                NaN              NaN                         NaN
mean           0.000507         0.000335                    0.006402
std            0.022518         0.018299                    0.083849
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                    4.000000

       AMT_REQ_CREDIT_BUREAU_DAY  AMT_REQ_CREDIT_BUREAU_WEEK  \
count               265992.000000               265992.000000
unique                        NaN                         NaN
top                           NaN                         NaN
freq                          NaN                         NaN
mean                     0.007000                    0.034362
std                      0.110757                    0.204685
min                      0.000000                    0.000000
25%                      0.000000                    0.000000
50%                      0.000000                    0.000000
75%                      0.000000                    0.000000
max                      9.000000                    8.000000

        AMT_REQ_CREDIT_BUREAU_MON  AMT_REQ_CREDIT_BUREAU_QRT  \
count                265992.000000               265992.000000
unique                         NaN                         NaN
top                            NaN                         NaN
freq                           NaN                         NaN
mean                      0.267395                    0.265474
std                       0.916002                    0.794056
min                       0.000000                    0.000000
25%                       0.000000                    0.000000
50%                       0.000000                    0.000000
75%                       0.000000                    0.000000
max                      27.000000                  261.000000

        AMT_REQ_CREDIT_BUREAU_YEAR
count                265992.000000
unique                         NaN
top                            NaN
freq                           NaN
```

```
mean                         1.899974
std                          1.869295
min                          0.000000
25%                          0.000000
50%                          1.000000
75%                          3.000000
max                         25.000000

[11 rows x 122 columns]
```

Missing data for application train

```
percent =
(datasets["application_train"].isnull().sum()/datasets["application_tr
ain"].isnull().count()*100).sort_values(ascending = False).round(2)
sum_missing =
datasets["application_train"].isna().sum().sort_values(ascending =
False)
missing_application_train_data  = pd.concat([percent, sum_missing],
axis=1, keys=['Percent', "Train Missing Count"])
missing_application_train_data.head(40)
```

	Percent	Train Missing Count
COMMONAREA_MEDI	69.87	214865
COMMONAREA_AVG	69.87	214865
COMMONAREA_MODE	69.87	214865
NONLIVINGAPARTMENTS_MODE	69.43	213514
NONLIVINGAPARTMENTS_AVG	69.43	213514
NONLIVINGAPARTMENTS_MEDI	69.43	213514
FONDKAPREMONT_MODE	68.39	210295
LIVINGAPARTMENTS_MODE	68.35	210199
LIVINGAPARTMENTS_AVG	68.35	210199
LIVINGAPARTMENTS_MEDI	68.35	210199
FLOORSMIN_AVG	67.85	208642
FLOORSMIN_MODE	67.85	208642
FLOORSMIN_MEDI	67.85	208642
YEARS_BUILD_MEDI	66.50	204488
YEARS_BUILD_MODE	66.50	204488
YEARS_BUILD_AVG	66.50	204488
OWN_CAR_AGE	65.99	202929
LANDAREA_MEDI	59.38	182590
LANDAREA_MODE	59.38	182590
LANDAREA_AVG	59.38	182590
BASEMENTAREA_MEDI	58.52	179943
BASEMENTAREA_AVG	58.52	179943
BASEMENTAREA_MODE	58.52	179943
EXT_SOURCE_1	56.38	173378
NONLIVINGAREA_MODE	55.18	169682
NONLIVINGAREA_AVG	55.18	169682
NONLIVINGAREA_MEDI	55.18	169682
ELEVATORS_MEDI	53.30	163891

```
ELEVATORS_AVG               53.30               163891
ELEVATORS_MODE              53.30               163891
WALLSMATERIAL_MODE          50.84               156341
APARTMENTS_MEDI             50.75               156061
APARTMENTS_AVG              50.75               156061
APARTMENTS_MODE             50.75               156061
ENTRANCES_MEDI              50.35               154828
ENTRANCES_AVG               50.35               154828
ENTRANCES_MODE              50.35               154828
LIVINGAREA_AVG              50.19               154350
LIVINGAREA_MODE             50.19               154350
LIVINGAREA_MEDI             50.19               154350
```

```
percent =
(datasets["application_test"].isnull().sum()/datasets["application_tes
t"].isnull().count()*100).sort_values(ascending = False).round(2)
sum_missing =
datasets["application_test"].isna().sum().sort_values(ascending =
False)
missing_application_train_data  = pd.concat([percent, sum_missing],
axis=1, keys=['Percent', "Test Missing Count"])
missing_application_train_data.head(20)
```

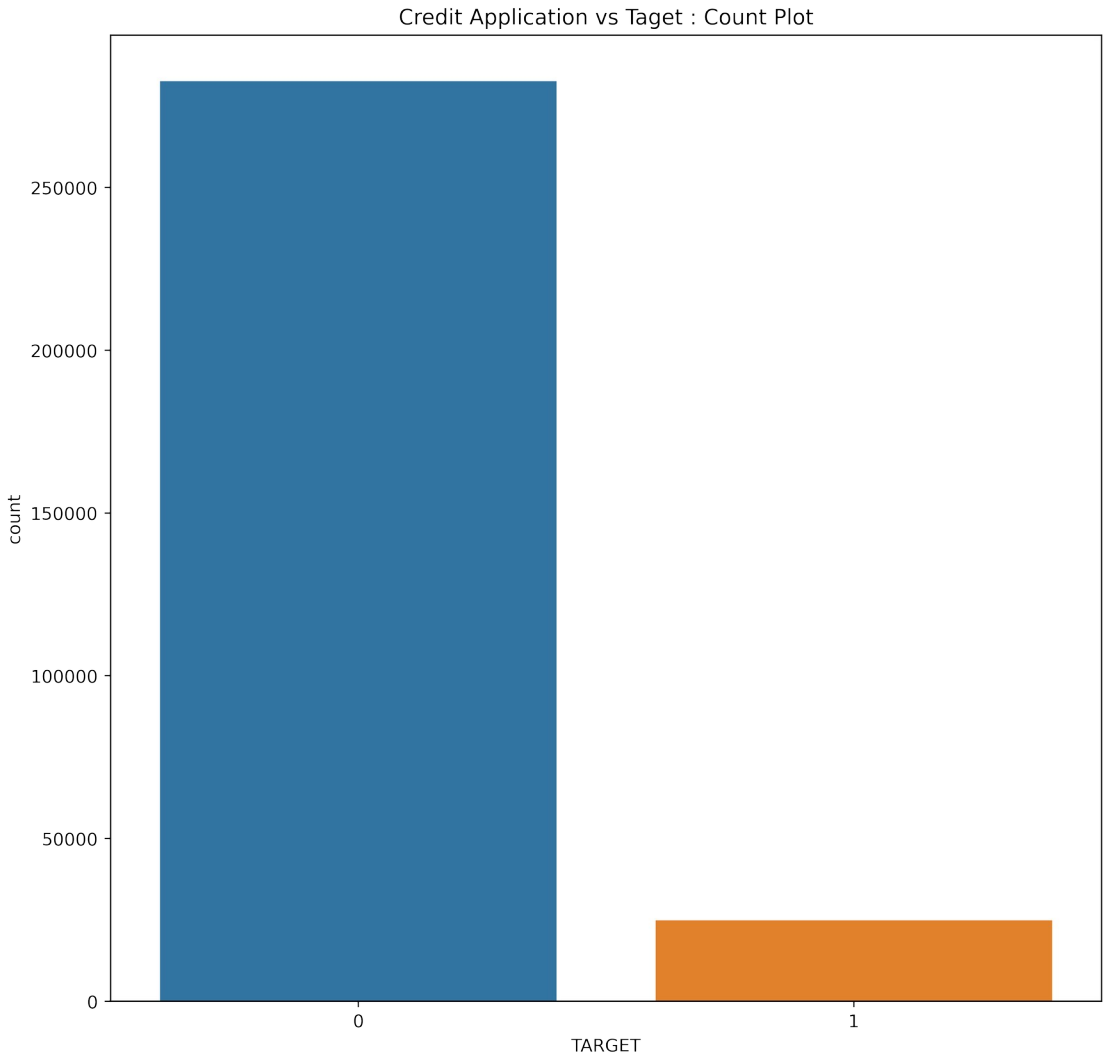
```
                           Percent  Test Missing Count
COMMONAREA_AVG               68.72               33495
COMMONAREA_MODE              68.72               33495
COMMONAREA_MEDI              68.72               33495
NONLIVINGAPARTMENTS_AVG      68.41               33347
NONLIVINGAPARTMENTS_MODE     68.41               33347
NONLIVINGAPARTMENTS_MEDI     68.41               33347
FONDKAPREMONT_MODE           67.28               32797
LIVINGAPARTMENTS_AVG         67.25               32780
LIVINGAPARTMENTS_MODE        67.25               32780
LIVINGAPARTMENTS_MEDI        67.25               32780
FLOORSMIN_MEDI               66.61               32466
FLOORSMIN_AVG                66.61               32466
FLOORSMIN_MODE               66.61               32466
OWN_CAR_AGE                  66.29               32312
YEARS_BUILD_AVG              65.28               31818
YEARS_BUILD_MEDI             65.28               31818
YEARS_BUILD_MODE             65.28               31818
LANDAREA_MEDI                57.96               28254
LANDAREA_AVG                 57.96               28254
LANDAREA_MODE                57.96               28254
```

Distribution of the target column

```
# datasets["application_train"]['TARGET'].astype(int).plot.hist();

fig, ax = plt.subplots(1,1, figsize=(10,10), dpi=400)
```

```
ax.set_title("Credit Application vs Taget : Count Plot")
sns.countplot(datasets["application_train"]['TARGET'], ax=ax)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fae2c9cf810>
```



Correlation with the target column

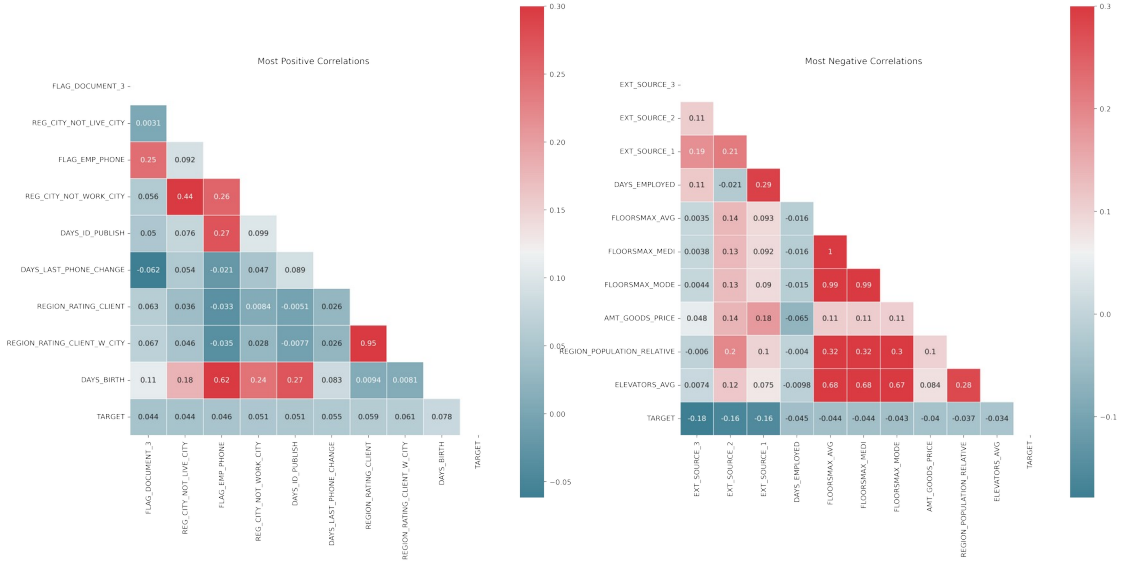
```
application_train_corr = datasets["application_train"].corr()
sorted_target = application_train_corr["TARGET"].sort_values()
tail_10 = sorted_target.tail(10)
head_10 = sorted_target.head(10)

print('Most Positive Correlations:\n', tail_10)
print('\nMost Negative Correlations:\n', head_10)
```

```
Most Positive Correlations:
 FLAG_DOCUMENT_3               0.044346
REG_CITY_NOT_LIVE_CITY         0.044395
FLAG_EMP_PHONE                 0.045982
```

```
REG_CITY_NOT_WORK_CITY         0.050994
DAYS_ID_PUBLISH                0.051457
DAYS_LAST_PHONE_CHANGE         0.055218
REGION_RATING_CLIENT           0.058899
REGION_RATING_CLIENT_W_CITY    0.060893
DAYS_BIRTH                     0.078239
TARGET                         1.000000
Name: TARGET, dtype: float64

Most Negative Correlations:
 EXT_SOURCE_3                 -0.178919
EXT_SOURCE_2                 -0.160472
EXT_SOURCE_1                 -0.155317
DAYS_EMPLOYED                -0.044932
FLOORSMAX_AVG                -0.044003
FLOORSMAX_MEDI               -0.043768
FLOORSMAX_MODE               -0.043226
AMT_GOODS_PRICE              -0.039645
REGION_POPULATION_RELATIVE   -0.037227
ELEVATORS_AVG                -0.034199
Name: TARGET, dtype: float64
```

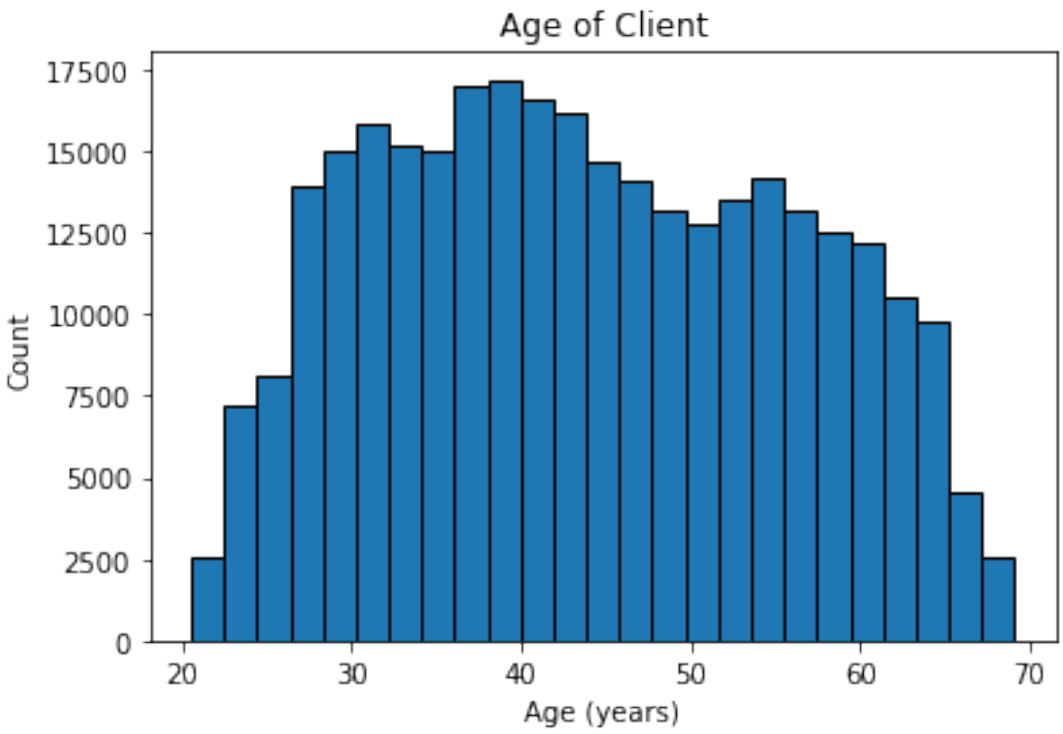
```
tail_10_corr = application_train_corr[[_ for _ in
tail_10.index]].loc[[_ for _ in tail_10.index]]
head_10_corr_list = [_ for _ in head_10.index]
head_10_corr_list.append("TARGET")
head_10_corr =
application_train_corr[head_10_corr_list].loc[head_10_corr_list]


# set up the matplotlib figure
f, ax = plt.subplots(1,2, figsize=(25, 25), dpi=400)

# generate a mask for the lower triangle
mask = np.zeros_like(tail_10_corr, dtype=np.bool_)
mask[np.triu_indices_from(mask)] = True

# generate a custom diverging colormap
cmap = sns.diverging_palette(220, 11, as_cmap=True)

# draw the heatmap with the mask and correct aspect ratio
sns.heatmap(tail_10_corr, mask=mask, cmap=cmap, vmax=.3,
            square=True,
            linewidths=.5, cbar_kws={"shrink": .5}, ax=ax[0],
annot=True);
ax[0].set_title("Most Positive Correlations")

# generate a mask for the lower triangle
mask = np.zeros_like(head_10_corr, dtype=np.bool_)
mask[np.triu_indices_from(mask)] = True
```

```
# generate a custom diverging colormap
cmap = sns.diverging_palette(220, 11, as_cmap=True)

# draw the heatmap with the mask and correct aspect ratio
sns.heatmap(head_10_corr, mask=mask, cmap=cmap, vmax=.3,
            square=True,
            linewidths=.5, cbar_kws={"shrink": .5}, ax=ax[1],
annot=True);

ax[1].set_title("Most Negative Correlations")


Text(0.5, 1.0, 'Most Negative Correlations')
```



Applicants Age

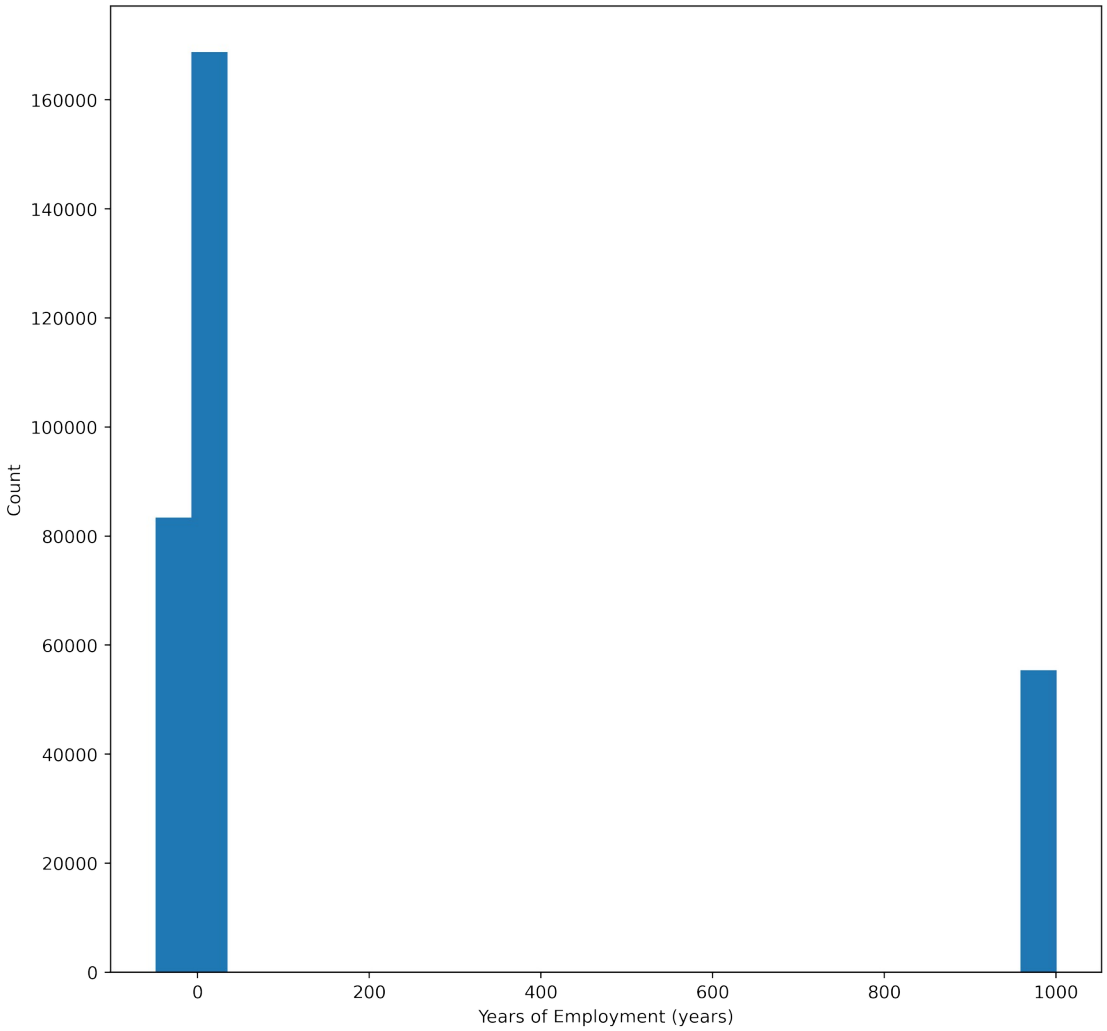
```
plt.hist(datasets["application_train"]['DAYS_BIRTH'] / -365, edgecolor
= 'k', bins = 25)
plt.title('Age of Client')
plt.xlabel('Age (years)')
plt.ylabel('Count')

Text(0, 0.5, 'Count')
```



Applicants years of employments

```
fig, ax = plt.subplots(1,1, figsize=(10,10),dpi=400)
plt.hist(datasets["application_train"]['DAYS_EMPLOYED'] /365, bins =
25)
plt.title('')
plt.xlabel('Years of Employment (years)')
plt.ylabel('Count')
```

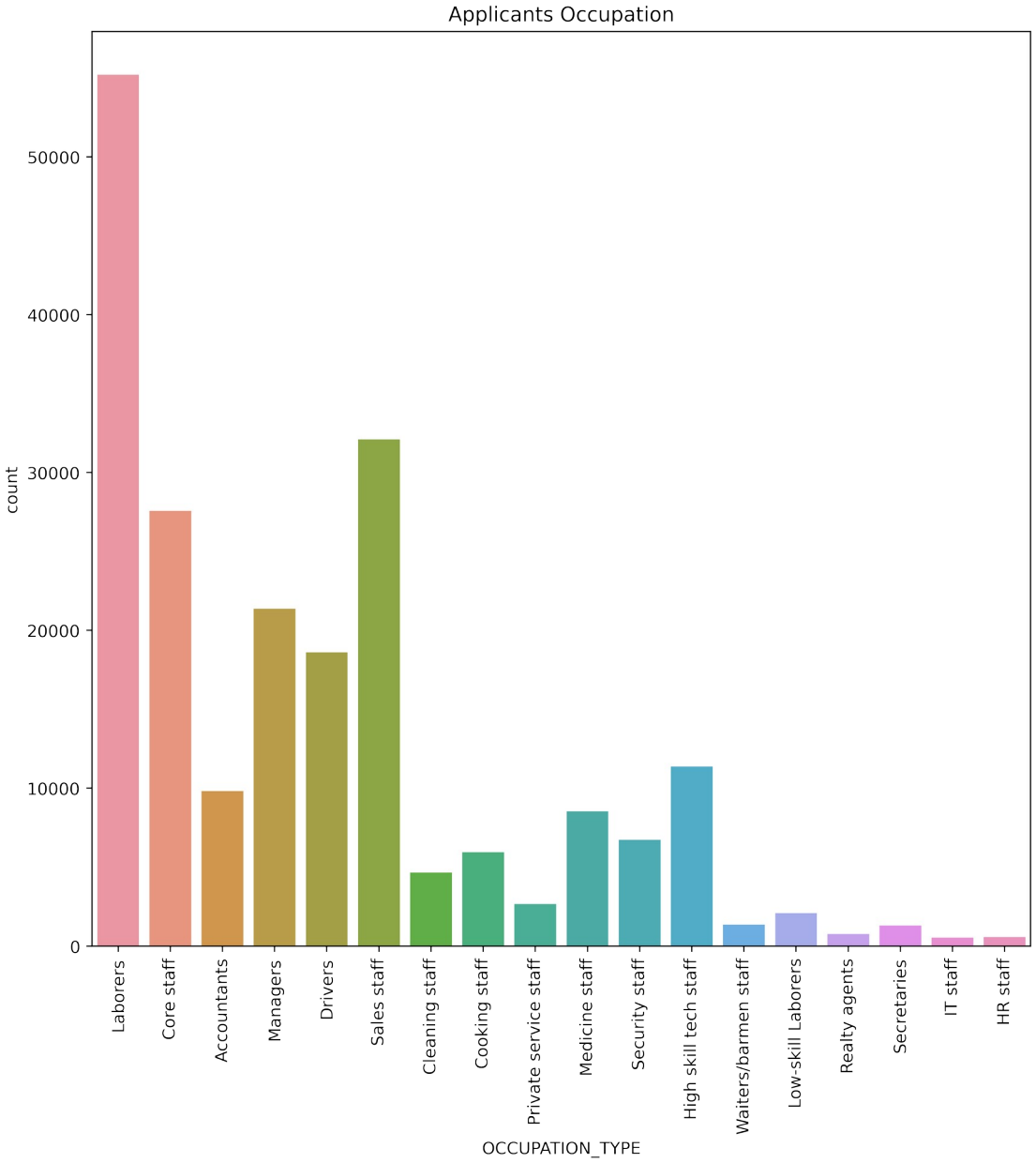
```
Text(0, 0.5, 'Count')
```



Interesting observation: DAYS_EMPLOYED : some rows have value as 365243(equivalent to 1000 years),i.e some people are employed for 1000 years

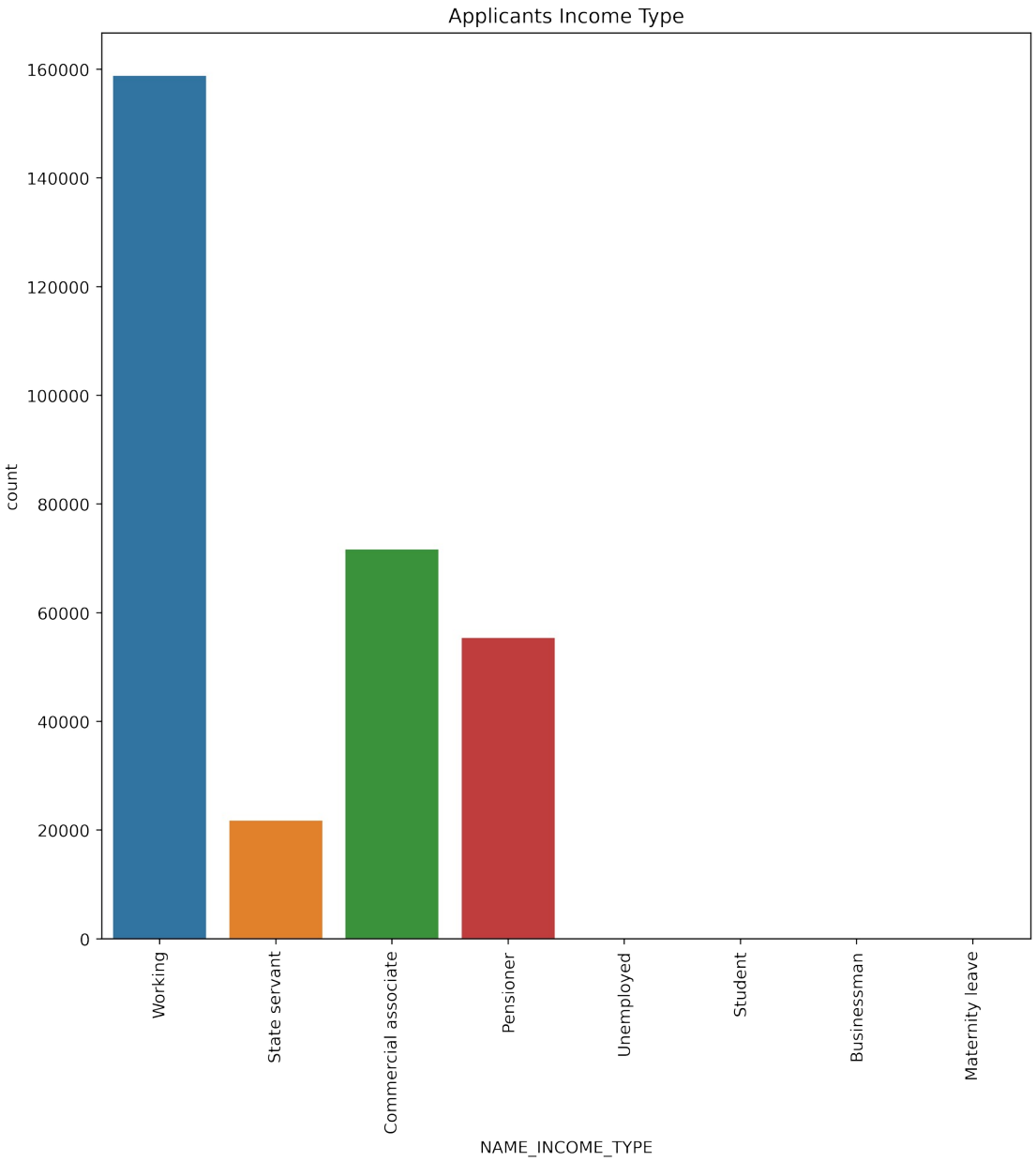
Applicants occupations

```
fig, ax =  plt.subplots(1,1, figsize=(10,10), dpi=400)
sns.countplot(x='OCCUPATION_TYPE',
data=datasets["application_train"],ax=ax);

ax.set_title('Applicants Occupation');
plt.xticks(rotation=90);
```

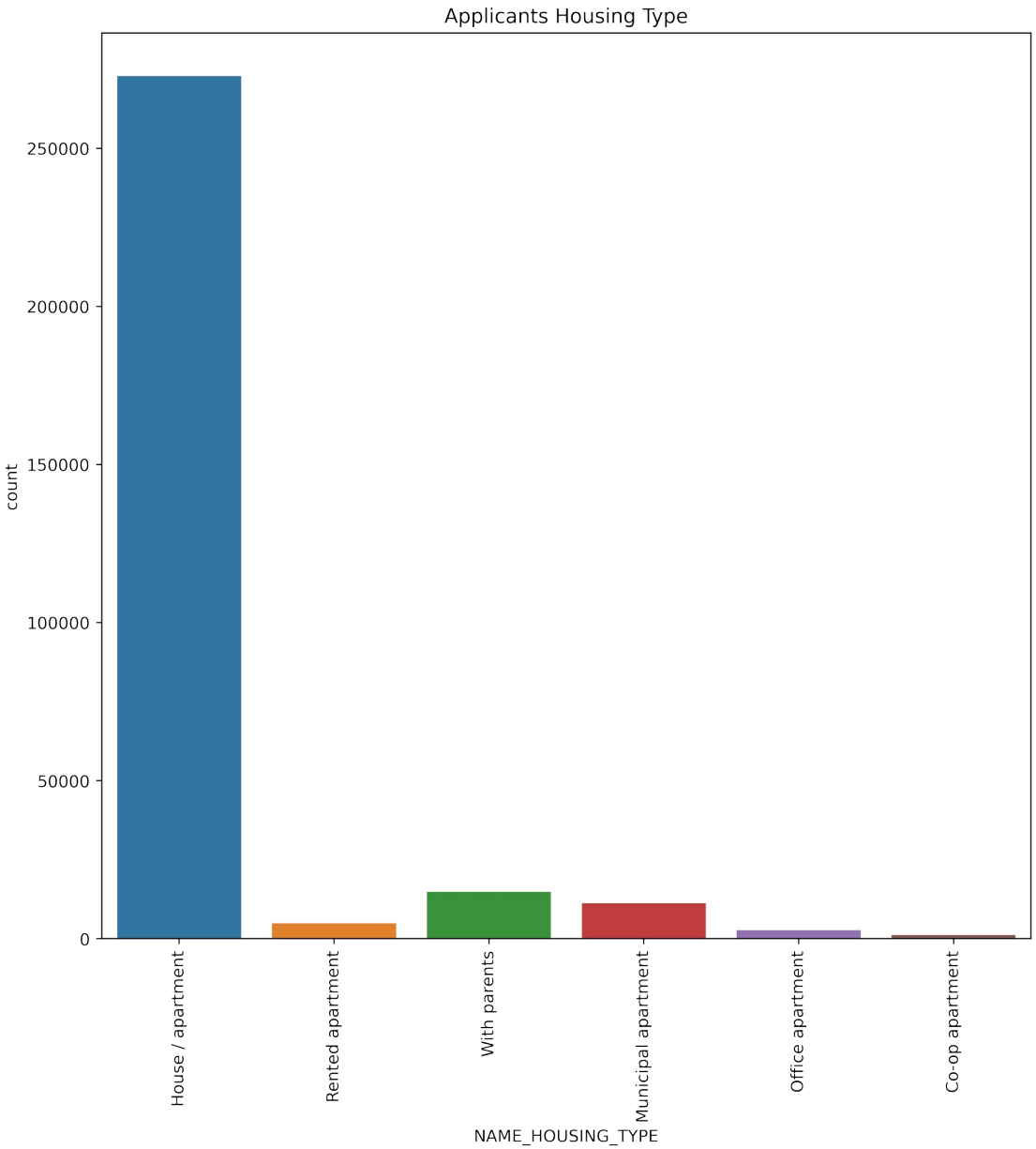


Applicants Income Type

```
fig, ax =  plt.subplots(1,1, figsize=(10,10), dpi=400)
sns.countplot(x='NAME_INCOME_TYPE',
data=datasets["application_train"],ax=ax);

ax.set_title('Applicants Income Type');
plt.xticks(rotation=90);
```

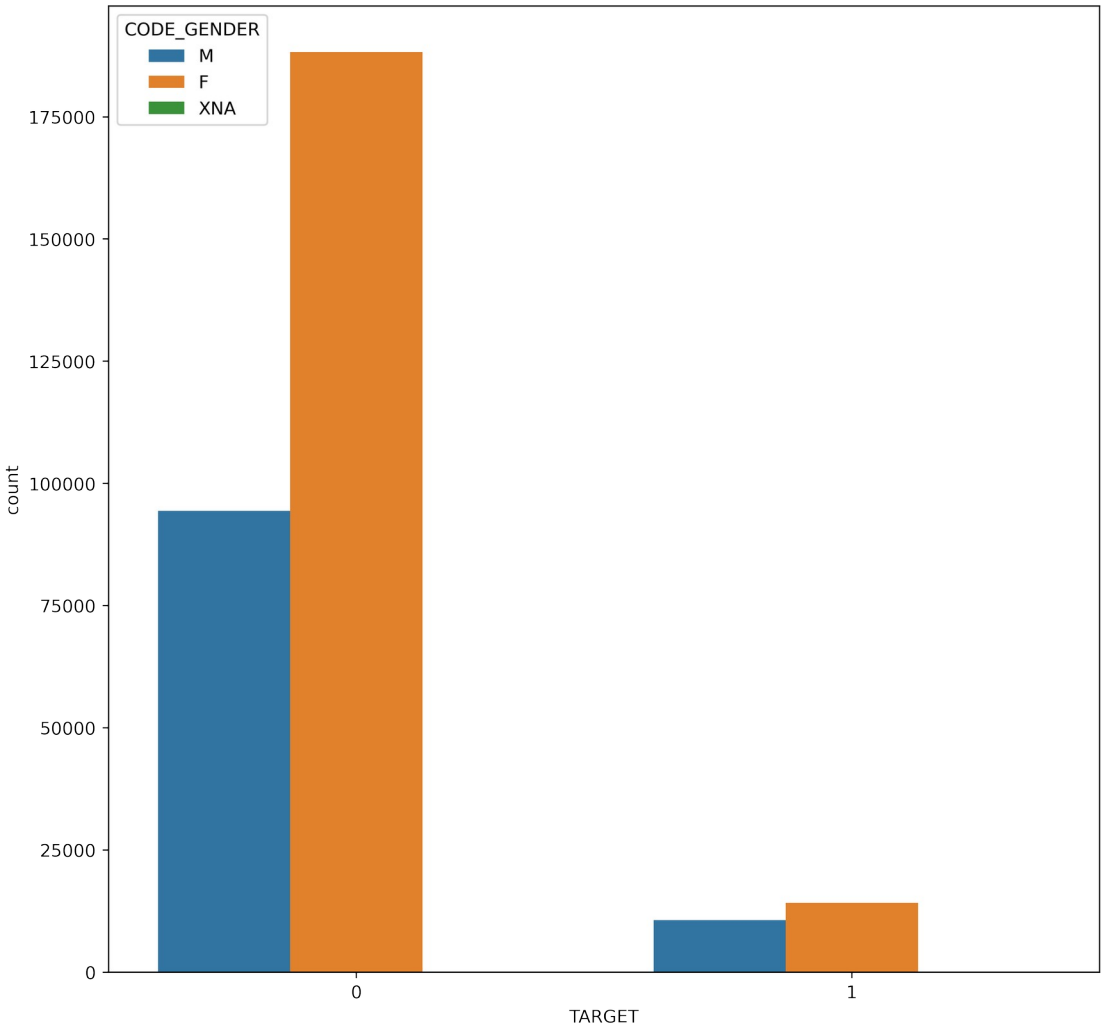



```
fig, ax =  plt.subplots(1,1, figsize=(10,10), dpi=400)
sns.countplot(x='NAME_HOUSING_TYPE',
data=datasets["application_train"],ax=ax);

ax.set_title('Applicants Housing Type');
plt.xticks(rotation=90);
```



Target vs Gender

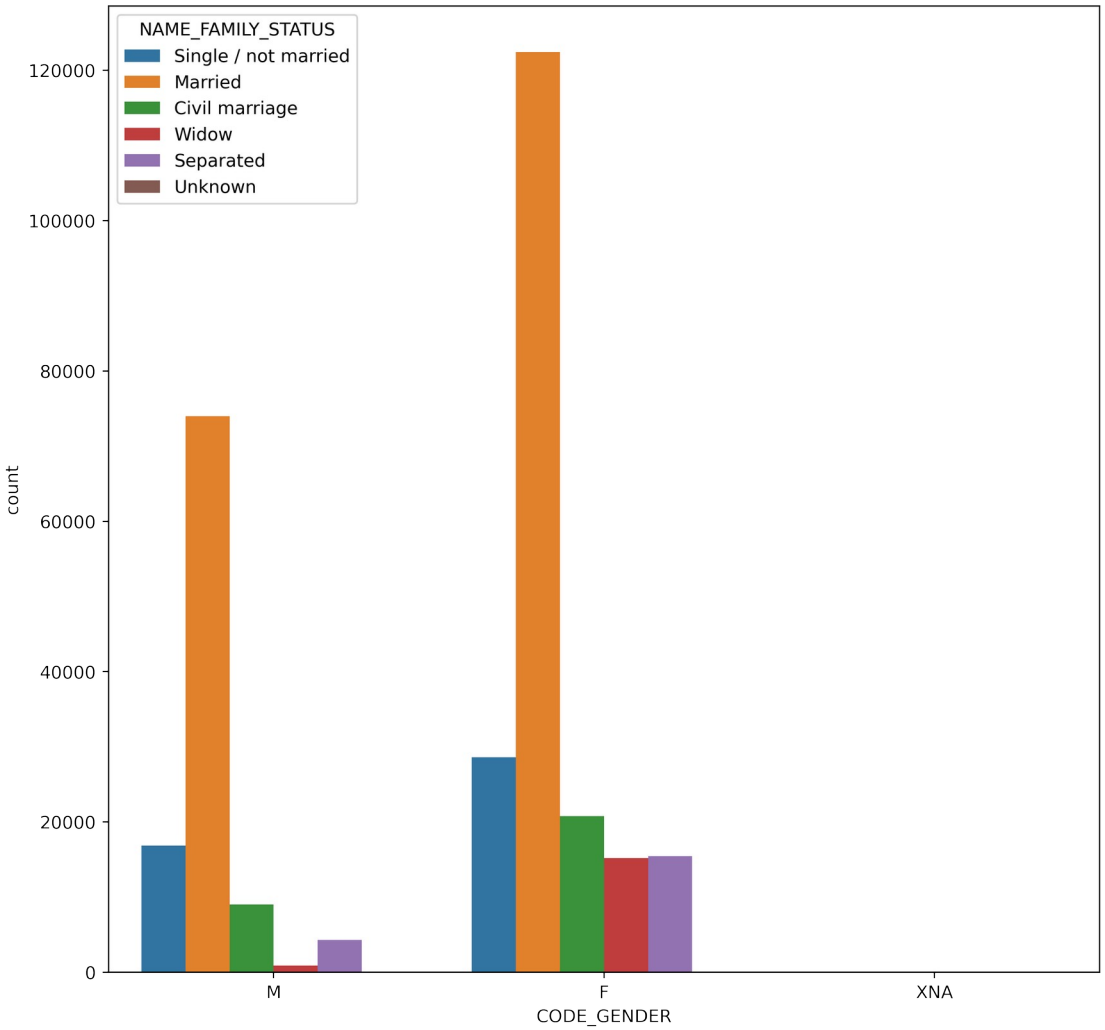
```
fig, ax =  plt.subplots(1,1, figsize=(10,10), dpi=400)
sns.countplot(x='TARGET', data=datasets["application_train"],ax=ax,
hue="CODE_GENDER");
```



Family Status of Loan applicants

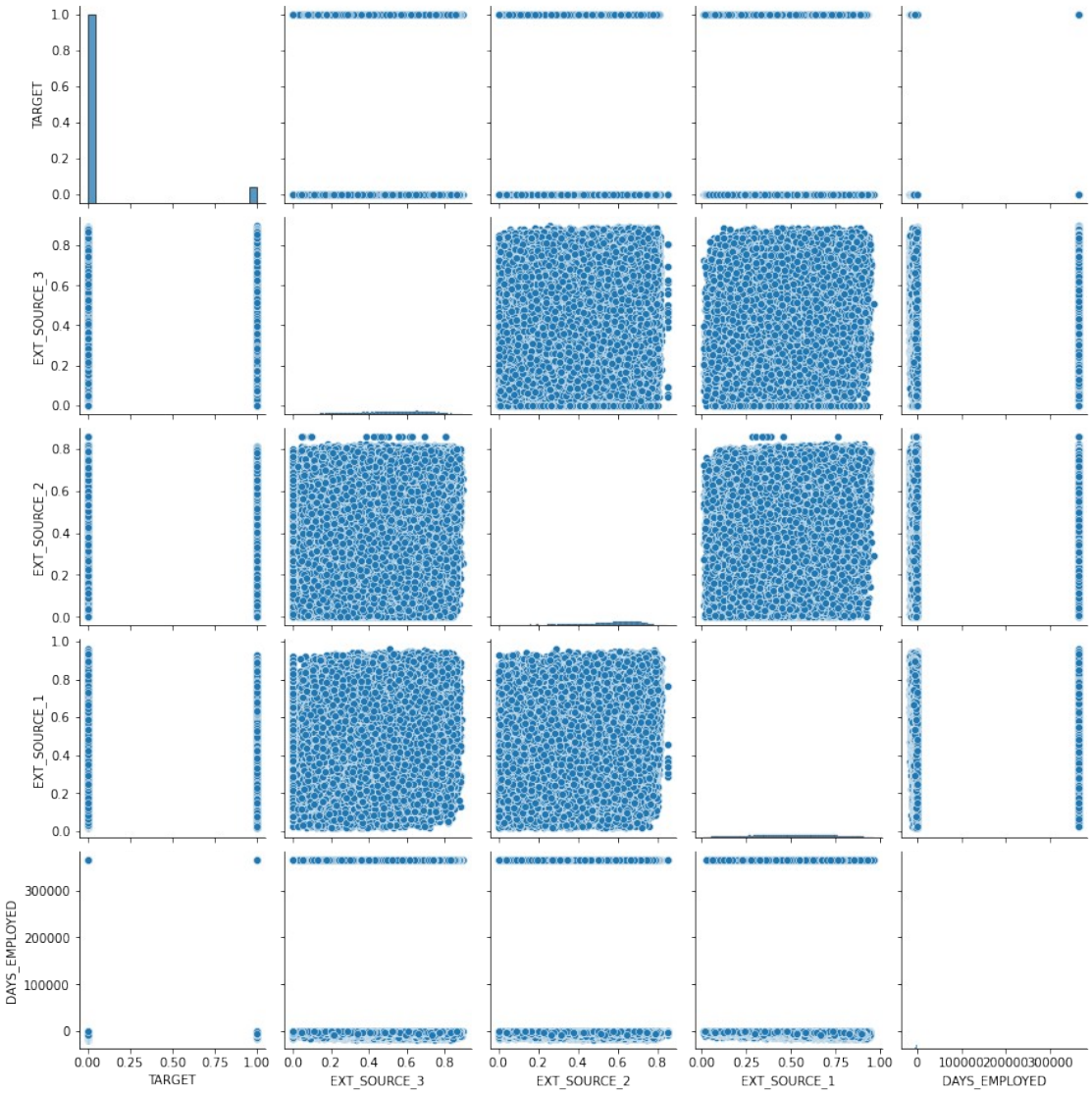
```
fig, ax =  plt.subplots(1,1, figsize=(10,10), dpi=400)
sns.countplot(x='CODE_GENDER',
data=datasets["application_train"],ax=ax, hue="NAME_FAMILY_STATUS")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fae27e693d0>
```




```
features = ["TARGET", "EXT_SOURCE_3", "EXT_SOURCE_2", "EXT_SOURCE_1",
"DAYS_EMPLOYED"]
sns.pairplot(datasets["application_train"][features])
```

```
<seaborn.axisgrid.PairGrid at 0x7fae27e23cd0>
```



Observation: The relationship between TARGET and "EXT_SOURCE_3", "EXT_SOURCE_2", "EXT_SOURCE_1", "DAYS_EMPLOYED" is not linear and monotonic.

Dataset : `bureau`

```
bur = datasets["bureau"]
```

```
bur.head(30)
```

```
    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
5 215354 5714467 Active currency 1 -
273
6 215354 5714468 Active currency 1 -
43
7 162297 5714469 Closed currency 1 -
1896
8 162297 5714470 Closed currency 1 -
1146
9 162297 5714471 Active currency 1 -
1146
10 162297 5714472 Active currency 1 -
1146
11 162297 5714473 Closed currency 1 -
2456
12 162297 5714474 Active currency 1 -
277
13 402440 5714475 Active currency 1 -
96
14 238881 5714482 Closed currency 1 -
318
15 238881 5714484 Closed currency 1 -
2911
16 238881 5714485 Closed currency 1 -
2148
17 238881 5714486 Active currency 1 -
381
18 238881 5714487 Active currency 1 -
95
19 238881 5714488 Closed currency 1 -
444
20 238881 5714489 Active currency 1 -
392
21 222183 5714491 Active currency 1 -
784
22 222183 5714492 Active currency 1 -
774
23 222183 5714493 Active currency 1 -
395
24 222183 5714495 Closed currency 1 -
2744
25 222183 5714496 Closed currency 1 -
1103
26 222183 5714497 Active currency 1 -
315
27 426155 5714498 Closed currency 1 -

```
1331
28      426155       5714499        Closed      currency 1        -
2534
29      426155       5714500        Closed      currency 1         -
845

    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
5                    0              27460.0                NaN
6                    0                 79.0                NaN
7                    0              -1684.0            -1710.0
8                    0               -811.0             -840.0
9                    0               -484.0                NaN
10                   0               -180.0                NaN
11                   0               -629.0             -825.0
12                   0               5261.0                NaN
13                   0                269.0                NaN
14                   0               -187.0             -187.0
15                   0              -2607.0            -2604.0
16                   0              -1595.0             -987.0
17                   0                  NaN                NaN
18                   0               1720.0                NaN
19                   0                -77.0              -77.0
20                   0                  NaN                NaN
21                   0               1008.0                NaN
22                   0                625.0                NaN
23                   0               1431.0                NaN
24                   0              -2561.0            -2559.0
25                   0                 -7.0             -343.0
26                   0               1512.0                NaN
27                   0               -994.0            -1023.0
28                   0              -2352.0            -2347.0
29                   0               -480.0             -480.0

    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                 77674.50                   0      2700000.00
5                     0.00                   0       180000.00
6                     0.00                   0        42103.80
7                 14985.00                   0        76878.45
8                     0.00                   0       103007.70
9                     0.00                   0         4500.00
10                    0.00                   0       337500.00
```

11	NaN	0	675000.00
12	0.00	0	7033500.00
13	0.00	0	89910.00
14	NaN	0	0.00
15	NaN	0	48555.00
16	NaN	0	135000.00
17	NaN	0	450000.00
18	NaN	0	67500.00
19	0.00	0	107184.06
20	0.00	0	252000.00
21	0.00	0	0.00
22	NaN	0	127840.50
23	NaN	0	1350000.00
24	310.50	0	18157.50
25	20493.27	0	675000.00
26	88821.00	0	3709552.50
27	1350.00	0	39433.50
28	NaN	0	38830.50
29	0.00	0	67500.00

	AMT_CREDIT_SUM_DEBT	AMT_CREDIT_SUM_LIMIT	AMT_CREDIT_SUM_OVERDUE
0	0.000	NaN	0.0
1	171342.000	NaN	0.0
2	NaN	NaN	0.0
3	NaN	NaN	0.0
4	NaN	NaN	0.0
5	71017.380	108982.620	0.0
6	42103.800	0.000	0.0
7	0.000	0.000	0.0
8	0.000	0.000	0.0
9	0.000	0.000	0.0
10	0.000	0.000	0.0
11	0.000	0.000	0.0
12	NaN	NaN	0.0
13	76905.000	0.000	0.0

14	0.000	0.000	0.0
15	NaN	NaN	0.0
16	NaN	NaN	0.0
17	520920.000	NaN	0.0
18	8131.500	NaN	0.0
19	0.000	0.000	0.0
20	23679.000	228320.100	0.0
21	-411.615	411.615	0.0
22	0.000	0.000	0.0
23	1185493.500	0.000	0.0
24	NaN	NaN	0.0
25	0.000	0.000	0.0
26	NaN	NaN	0.0
27	0.000	0.000	0.0
28	0.000	0.000	0.0
29	0.000	0.000	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
5	Credit card	-31	NaN
6	Consumer credit	-22	NaN
7	Consumer credit	-1710	NaN
8	Consumer credit	-840	NaN
9	Credit card	-690	NaN
10	Credit card	-690	NaN
11	Consumer credit	-706	NaN
12	Mortgage	-31	NaN

```
13  Consumer credit                -22          NaN
14      Credit card               -185          NaN
15  Consumer credit              -2601          NaN
16  Consumer credit               -984          NaN
17  Consumer credit                 -4          NaN
18      Credit card                 -7          NaN
19  Consumer credit                -71          NaN
20      Credit card                -22          NaN
21      Credit card               -694          NaN
22      Credit card               -210          NaN
23  Consumer credit                -24          NaN
24  Consumer credit              -2559          NaN
25  Consumer credit               -343          NaN
26         Car loan                -32          NaN
27  Consumer credit              -1023          NaN
28  Consumer credit              -2345          NaN
29  Consumer credit               -480          NaN
```

```
bur.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1716428 entries, 0 to 1716427
Data columns (total 17 columns):
 #   Column                  Dtype
---  ------                  -----
 0   SK_ID_CURR              int64
 1   SK_ID_BUREAU            int64
 2   CREDIT_ACTIVE           object
 3   CREDIT_CURRENCY         object
 4   DAYS_CREDIT             int64
 5   CREDIT_DAY_OVERDUE      int64
 6   DAYS_CREDIT_ENDDATE     float64
 7   DAYS_ENDDATE_FACT       float64
 8   AMT_CREDIT_MAX_OVERDUE  float64
 9   CNT_CREDIT_PROLONG      int64
 10  AMT_CREDIT_SUM          float64
 11  AMT_CREDIT_SUM_DEBT     float64
 12  AMT_CREDIT_SUM_LIMIT    float64
 13  AMT_CREDIT_SUM_OVERDUE  float64
 14  CREDIT_TYPE             object
 15  DAYS_CREDIT_UPDATE      int64
 16  AMT_ANNUITY             float64
dtypes: float64(8), int64(6), object(3)
memory usage: 222.6+ MB
```

```
bur.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')
```

```
bur.describe()
```

```
         SK_ID_CURR  SK_ID_BUREAU   DAYS_CREDIT  CREDIT_DAY_OVERDUE  \
count  1.716428e+06  1.716428e+06  1.716428e+06        1.716428e+06
mean   2.782149e+05  5.924434e+06 -1.142108e+03        8.181666e-01
std    1.029386e+05  5.322657e+05  7.951649e+02        3.654443e+01
min    1.000010e+05  5.000000e+06 -2.922000e+03        0.000000e+00
25%    1.888668e+05  5.463954e+06 -1.666000e+03        0.000000e+00
50%    2.780550e+05  5.926304e+06 -9.870000e+02        0.000000e+00
75%    3.674260e+05  6.385681e+06 -4.740000e+02        0.000000e+00
max    4.562550e+05  6.843457e+06  0.000000e+00        2.792000e+03

       DAYS_CREDIT_ENDDATE  DAYS_ENDDATE_FACT  AMT_CREDIT_MAX_OVERDUE
\
count         1.610875e+06       1.082775e+06            5.919400e+05

mean          5.105174e+02      -1.017437e+03            3.825418e+03

std           4.994220e+03       7.140106e+02            2.060316e+05

min          -4.206000e+04      -4.202300e+04            0.000000e+00

25%          -1.138000e+03      -1.489000e+03            0.000000e+00

50%          -3.300000e+02      -8.970000e+02            0.000000e+00

75%           4.740000e+02      -4.250000e+02            0.000000e+00

max           3.119900e+04       0.000000e+00            1.159872e+08


       CNT_CREDIT_PROLONG  AMT_CREDIT_SUM  AMT_CREDIT_SUM_DEBT  \
count        1.716428e+06    1.716415e+06         1.458759e+06
mean         6.410406e-03    3.549946e+05         1.370851e+05
std          9.622391e-02    1.149811e+06         6.774011e+05
min          0.000000e+00    0.000000e+00        -4.705600e+06
25%          0.000000e+00    5.130000e+04         0.000000e+00
50%          0.000000e+00    1.255185e+05         0.000000e+00
75%          0.000000e+00    3.150000e+05         4.015350e+04
max          9.000000e+00    5.850000e+08         1.701000e+08

       AMT_CREDIT_SUM_LIMIT  AMT_CREDIT_SUM_OVERDUE
```

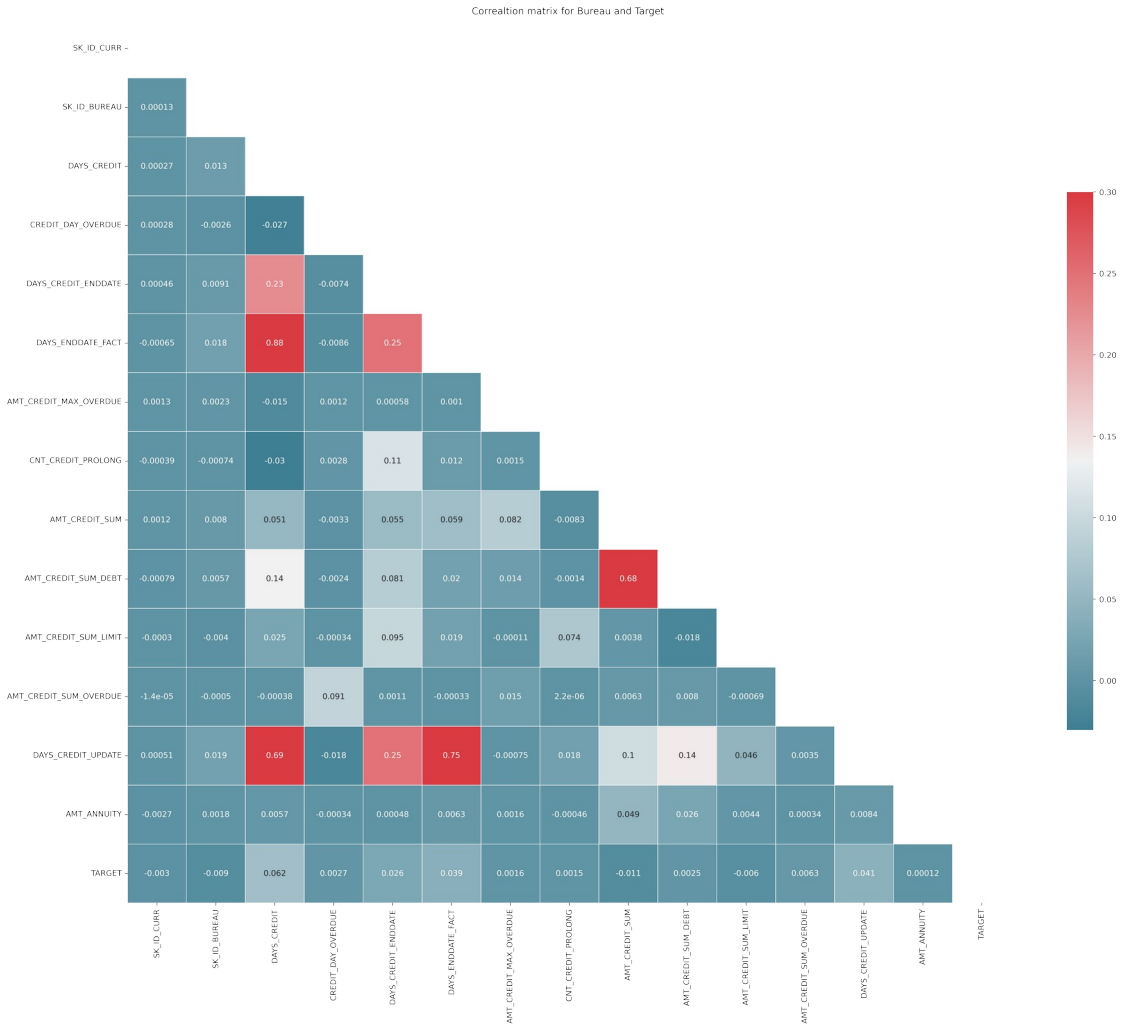
```
DAYS_CREDIT_UPDATE  \
count          1.124648e+06           1.716428e+06
1.716428e+06
mean           6.229515e+03           3.791276e+01       -
5.937483e+02
std            4.503203e+04           5.937650e+03
7.207473e+02
min           -5.864061e+05           0.000000e+00       -
4.194700e+04
25%            0.000000e+00           0.000000e+00       -
9.080000e+02
50%            0.000000e+00           0.000000e+00       -
3.950000e+02
75%            0.000000e+00           0.000000e+00       -
3.300000e+01
max            4.705600e+06           3.756681e+06
3.720000e+02

        AMT_ANNUITY
count  4.896370e+05
mean   1.571276e+04
std    3.258269e+05
min    0.000000e+00
25%    0.000000e+00
50%    0.000000e+00
75%    1.350000e+04
max    1.184534e+08

bur.isna().sum()

SK_ID_CURR                      0
SK_ID_BUREAU                    0
CREDIT_ACTIVE                   0
CREDIT_CURRENCY                 0
DAYS_CREDIT                     0
CREDIT_DAY_OVERDUE              0
DAYS_CREDIT_ENDDATE        105553
DAYS_ENDDATE_FACT          633653
AMT_CREDIT_MAX_OVERDUE    1124488
CNT_CREDIT_PROLONG              0
AMT_CREDIT_SUM                 13
AMT_CREDIT_SUM_DEBT        257669
AMT_CREDIT_SUM_LIMIT       591780
AMT_CREDIT_SUM_OVERDUE          0
CREDIT_TYPE                     0
DAYS_CREDIT_UPDATE              0
AMT_ANNUITY               1226791
dtype: int64
```

```
app_train = datasets['application_train']
app_train_req = app_train[["SK_ID_CURR", "TARGET"]]
merged_df = pd.merge(bur, app_train_req, how="left")

merged_df.corr()["TARGET"].sort_values()
```

```
AMT_CREDIT_SUM           -0.010606
SK_ID_BUREAU             -0.009018
AMT_CREDIT_SUM_LIMIT     -0.005990
SK_ID_CURR               -0.003024
AMT_ANNUITY               0.000117
CNT_CREDIT_PROLONG        0.001523
AMT_CREDIT_MAX_OVERDUE    0.001587
AMT_CREDIT_SUM_DEBT       0.002539
CREDIT_DAY_OVERDUE        0.002652
AMT_CREDIT_SUM_OVERDUE    0.006253
DAYS_CREDIT_ENDDATE       0.026497
DAYS_ENDDATE_FACT         0.039057
DAYS_CREDIT_UPDATE        0.041076
DAYS_CREDIT               0.061556
TARGET                    1.000000
Name: TARGET, dtype: float64
```

```
# set up the matplotlib figure
f, ax = plt.subplots(1,1, figsize=(25, 25),dpi=400)

# generate a mask for the lower triangle
mask = np.zeros_like(merged_df.corr(), dtype=np.bool_)
mask[np.triu_indices_from(mask)] = True

# generate a custom diverging colormap
cmap = sns.diverging_palette(220, 11, as_cmap=True)

# draw the heatmap with the mask and correct aspect ratio
sns.heatmap(merged_df.corr(), mask=mask, cmap=cmap, vmax=.3,
            square=True,
            linewidths=.5, cbar_kws={"shrink": .5}, ax=ax,
annot=True);
ax.set_title("Correaltion matrix for Bureau and Target ")
```

```
Text(0.5, 1.0, 'Correaltion matrix for Bureau and Target ')
```




```
app_train_req[app_train_req.TARGET==0]
```

```
        SK_ID_CURR  TARGET
1           100003       0
2           100004       0
3           100006       0
4           100007       0
5           100008       0
...            ...     ...
307505      456249       0
307506      456251       0
307507      456252       0
307508      456253       0
307510      456255       0

[282686 rows x 2 columns]
```

```
len(app_train_req.SK_ID_CURR.unique())
```

```
307511
```

```
len(bur.SK_ID_CURR.unique())
```

```
305811
```

```
len(datasets["application_test"].SK_ID_CURR.unique())
```

```
48744
```

```
train_diff = np.setdiff1d(bur.SK_ID_CURR.unique(),
app_train_req.SK_ID_CURR.unique())
test_diff = np.setdiff1d(train_diff,
datasets["application_test"].SK_ID_CURR.unique())
len(test_diff)
```

```
0
```

Observation: Bureau contains record for all SK_ID_CURR

```
observation_ids = [100002, 100031, 100003]
for grp, df in bur.groupby("SK_ID_CURR"):
    if grp in observation_ids:
        display(pd.merge(df, app_train_req, how="left"))
        observation_ids.remove(grp)
        if len(observation_ids) ==0:
            break
if len(observation_ids) !=0:
    print("NO PREVIOUS APPLICATION FOUND FOR :
{}".format(observation_ids))
```

```
   SK_ID_CURR  SK_ID_BUREAU CREDIT_ACTIVE CREDIT_CURRENCY  DAYS_CREDIT
\
0      100002       6158904        Closed      currency 1        -1125

1      100002       6158905        Closed      currency 1         -476

2      100002       6158906        Closed      currency 1        -1437

3      100002       6158907        Closed      currency 1        -1121

4      100002       6158908        Closed      currency 1         -645

5      100002       6158909        Active      currency 1         -103

6      100002       6158903        Active      currency 1        -1042

7      100002       6113835        Closed      currency 1        -1043


   CREDIT_DAY_OVERDUE  DAYS_CREDIT_ENDDATE  DAYS_ENDDATE_FACT  \
0                   0              -1038.0            -1038.0
1                   0                  NaN              -48.0
```

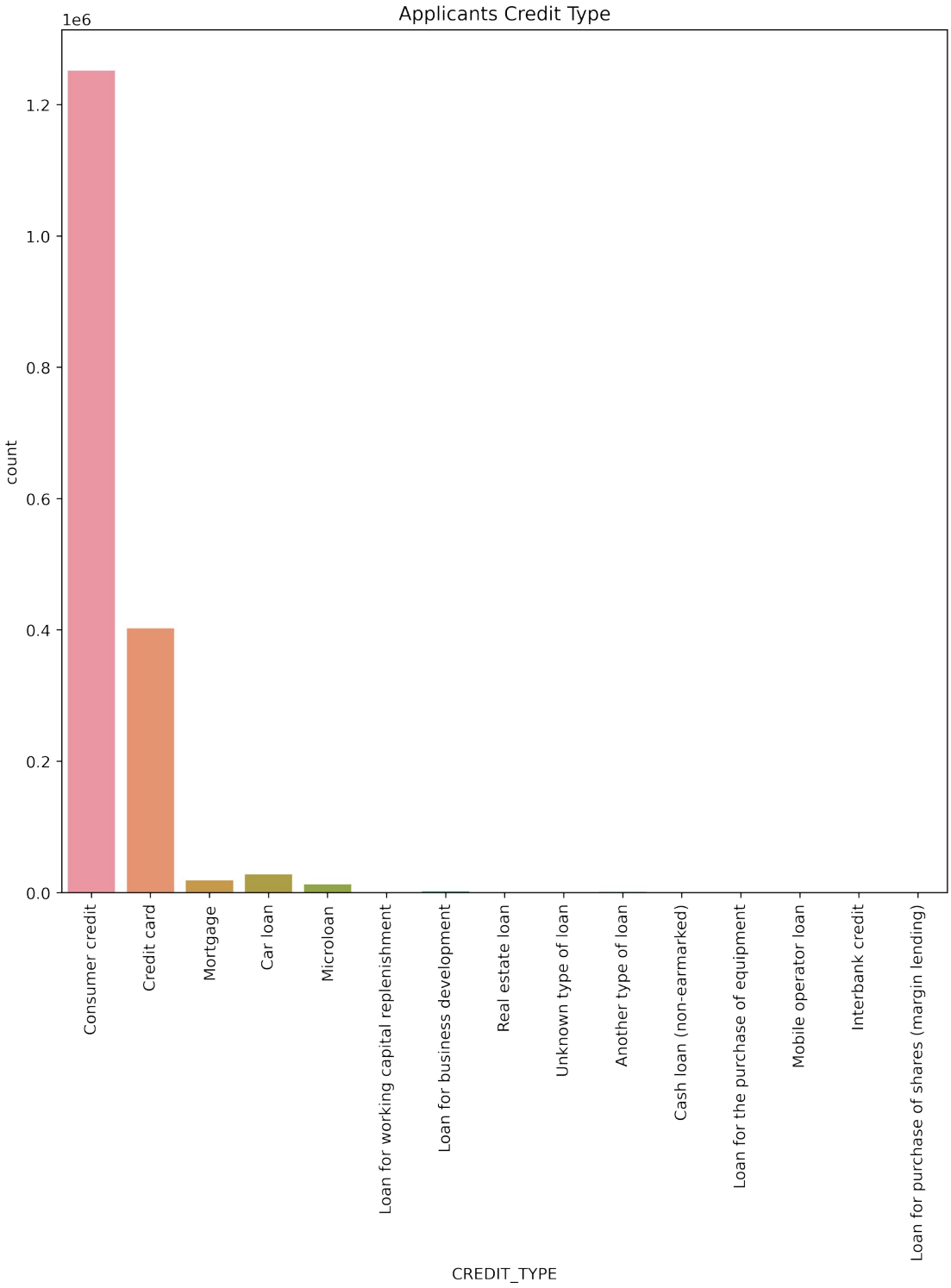
```
2                   0             -1072.0            -1185.0
3                   0              -911.0             -911.0
4                   0                85.0              -36.0
5                   0                 NaN                NaN
6                   0               780.0                NaN
7                   0                62.0             -967.0

   AMT_CREDIT_MAX_OVERDUE  CNT_CREDIT_PROLONG  AMT_CREDIT_SUM  \
0                     NaN                   0       40761.000
1                     NaN                   0           0.000
2                   0.000                   0      135000.000
3                3321.000                   0       19071.000
4                5043.645                   0      120735.000
5                  40.500                   0       31988.565
6                     NaN                   0      450000.000
7                   0.000                   0       67500.000

   AMT_CREDIT_SUM_DEBT  AMT_CREDIT_SUM_LIMIT
AMT_CREDIT_SUM_OVERDUE  \
0                  NaN                   NaN                     0.0

1                  0.0                   NaN                     0.0

2                  0.0                 0.000                     0.0

3                  NaN                   NaN                     0.0

4                  0.0                 0.000                     0.0

5                  0.0             31988.565                     0.0

6             245781.0                 0.000                     0.0

7                  NaN                   NaN                     0.0


       CREDIT_TYPE  DAYS_CREDIT_UPDATE  AMT_ANNUITY  TARGET
0      Credit card               -1038          0.0       1
1      Credit card                 -47          NaN       1
2  Consumer credit               -1185          0.0       1
3  Consumer credit                -906          0.0       1
4  Consumer credit                 -34          0.0       1
5      Credit card                 -24          0.0       1
6  Consumer credit                  -7          0.0       1
7      Credit card                -758          0.0       1

   SK_ID_CURR  SK_ID_BUREAU CREDIT_ACTIVE CREDIT_CURRENCY  DAYS_CREDIT
\
0      100003       5885877        Closed      currency 1        -2586
```

```
1      100003       5885878        Closed      currency 1        -1636

2      100003       5885879        Closed      currency 1         -775

3      100003       5885880        Active      currency 1         -606


   CREDIT_DAY_OVERDUE  DAYS_CREDIT_ENDDATE  DAYS_ENDDATE_FACT  \
0                   0              -2434.0            -2131.0
1                   0               -540.0             -540.0
2                   0               -420.0             -621.0
3                   0               1216.0                NaN

   AMT_CREDIT_MAX_OVERDUE  CNT_CREDIT_PROLONG  AMT_CREDIT_SUM  \
0                     0.0                   0         22248.0
1                     0.0                   0        112500.0
2                     0.0                   0         72652.5
3                     0.0                   0        810000.0

   AMT_CREDIT_SUM_DEBT  AMT_CREDIT_SUM_LIMIT
AMT_CREDIT_SUM_OVERDUE  \
0                  0.0                   0.0                      0.0

1                  0.0                   0.0                      0.0

2                  0.0                   0.0                      0.0

3                  0.0              810000.0                      0.0


       CREDIT_TYPE  DAYS_CREDIT_UPDATE  AMT_ANNUITY  TARGET
0  Consumer credit               -2131          NaN       0
1      Credit card                -540          NaN       0
2  Consumer credit                -550          NaN       0
3      Credit card                 -43          NaN       0

   SK_ID_CURR  SK_ID_BUREAU CREDIT_ACTIVE CREDIT_CURRENCY  DAYS_CREDIT
\
0      100031       6187667        Active      currency 1         -180

1      100031       6187661        Closed      currency 1        -1365

2      100031       6187662        Closed      currency 1         -992

3      100031       6187663        Closed      currency 1         -501

4      100031       6187664        Closed      currency 1         -203
```

```
5      100031       6187665        Active      currency 1         -741

6      100031       6187666        Active      currency 1          -75


   CREDIT_DAY_OVERDUE  DAYS_CREDIT_ENDDATE  DAYS_ENDDATE_FACT  \
0                   0                  NaN                NaN
1                   0               -269.0             -991.0
2                   0                469.0             -499.0
3                   0                960.0             -233.0
4                   0                162.0              -73.0
5                   0                  NaN                NaN
6                   0               1021.0                NaN

   AMT_CREDIT_MAX_OVERDUE  CNT_CREDIT_PROLONG  AMT_CREDIT_SUM  \
0                     NaN                   0         90000.0
1                     NaN                   0        573358.5
2                     NaN                   0       1125000.0
3                     NaN                   0       1350000.0
4                     NaN                   0        171013.5
5                     NaN                   0        112500.0
6                     NaN                   0        548779.5

   AMT_CREDIT_SUM_DEBT  AMT_CREDIT_SUM_LIMIT
AMT_CREDIT_SUM_OVERDUE  \
0                  NaN                   NaN                     0.0

1                  0.0                   NaN                     0.0

2            1125000.0                   NaN                     0.0

3                  NaN                   NaN                     0.0

4                  NaN                   NaN                     0.0

5                  NaN                   NaN                     0.0

6                  NaN                   NaN                     0.0


       CREDIT_TYPE  DAYS_CREDIT_UPDATE  AMT_ANNUITY  TARGET
0      Credit card                  -9          NaN       1
1  Consumer credit                -989          NaN       1
2  Consumer credit                -499          NaN       1
3  Consumer credit                -211          NaN       1
4  Consumer credit                 -72          NaN       1
5      Credit card                 -72          NaN       1
6  Consumer credit                 -12          NaN       1
```

```
fig, ax =  plt.subplots(1,1, figsize=(10,10),dpi=400)
sns.countplot(x='CREDIT_TYPE', data=bur,ax=ax)
ax.set_title("Applicants Credit Type")
plt.xticks(rotation="90")
```

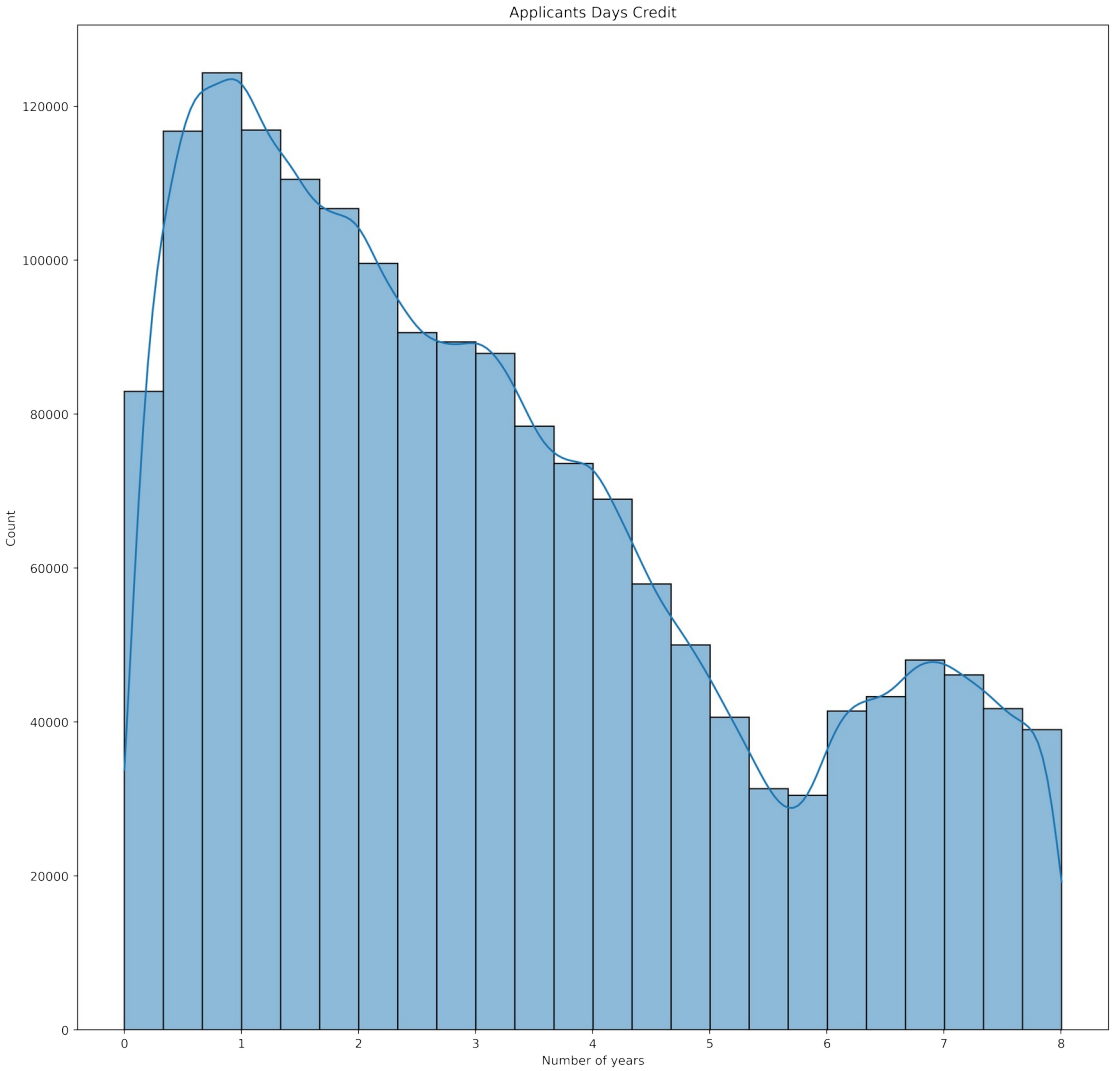
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14]),
 <a list of 15 Text major ticklabel objects>)
```



Bureau: Applicants Days Credit

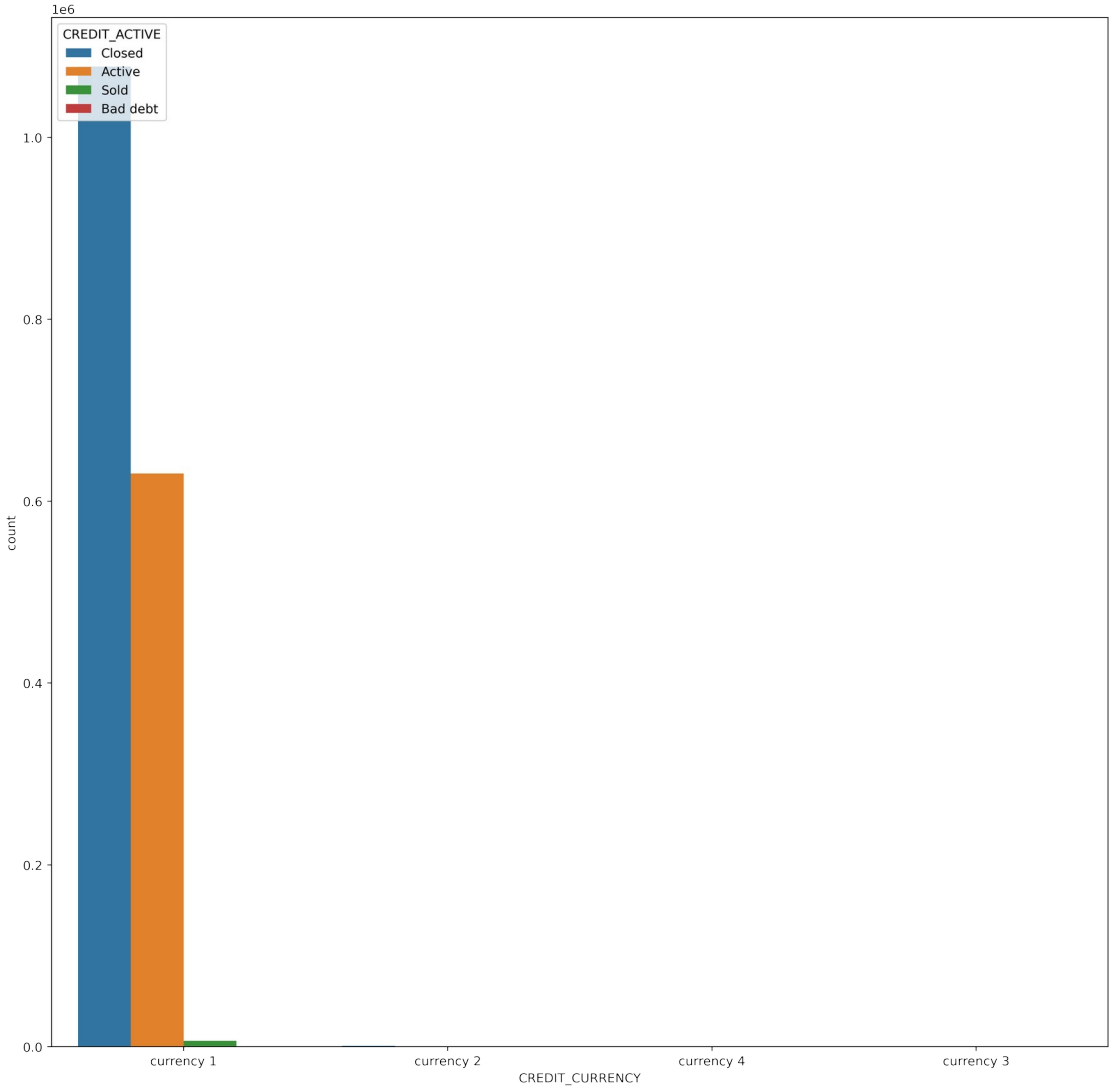
```
fig, ax =  plt.subplots(1,1, figsize=(15,15), dpi=400)
sns.histplot(bur["DAYS_CREDIT"]/-365,ax=ax, kde=True, bins=24)
ax.set_title("Applicants Days Credit")
ax.set_xlabel("Number of years")
```

```
Text(0.5, 0, 'Number of years')
```




```
fig, ax =  plt.subplots(1,1, figsize=(15,15), dpi=400)
sns.countplot(x="CREDIT_CURRENCY", data=bur,ax=ax,
hue="CREDIT_ACTIVE")
```

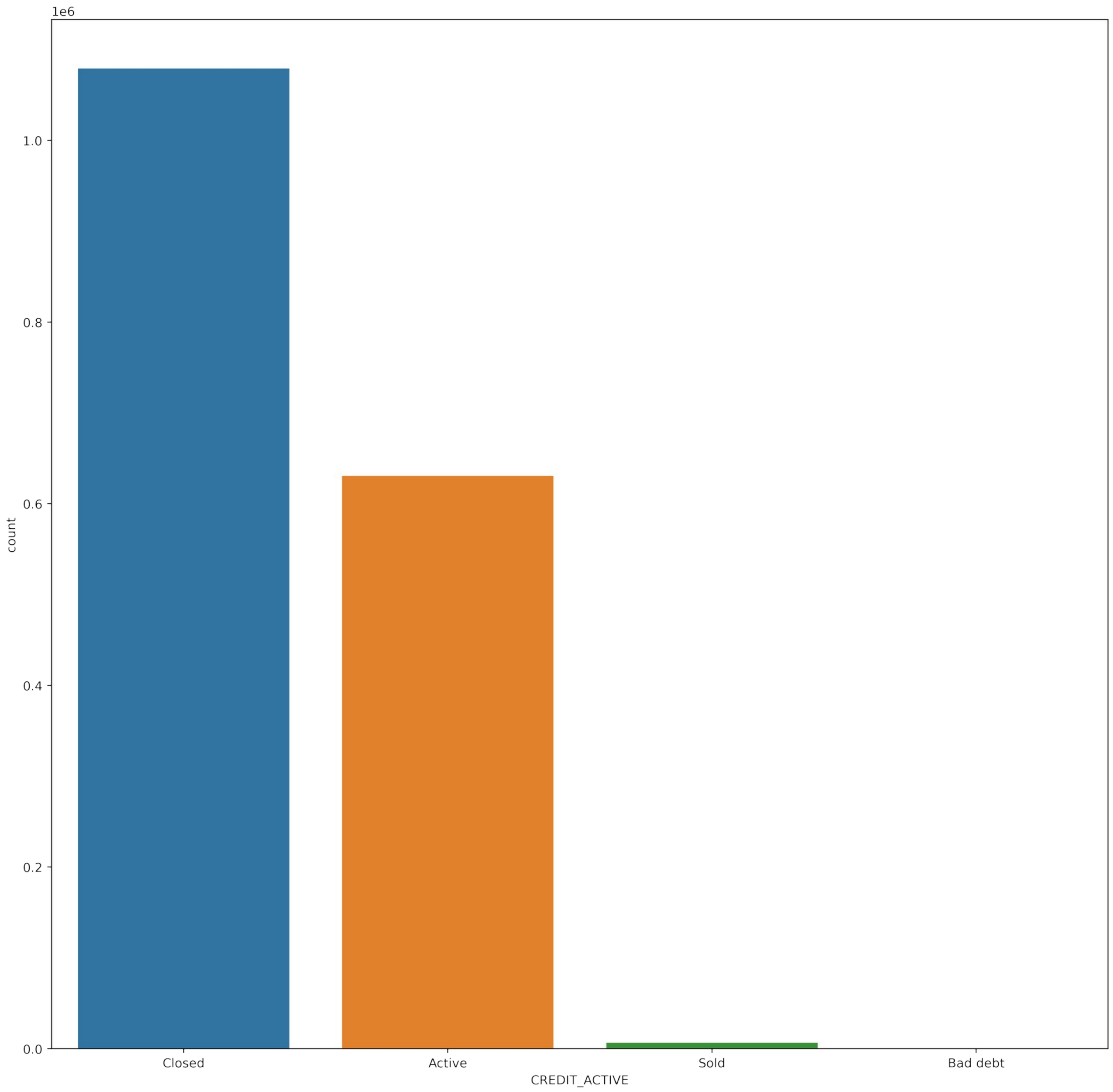
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fae1fb366d0>
```



Applicants Credit history status

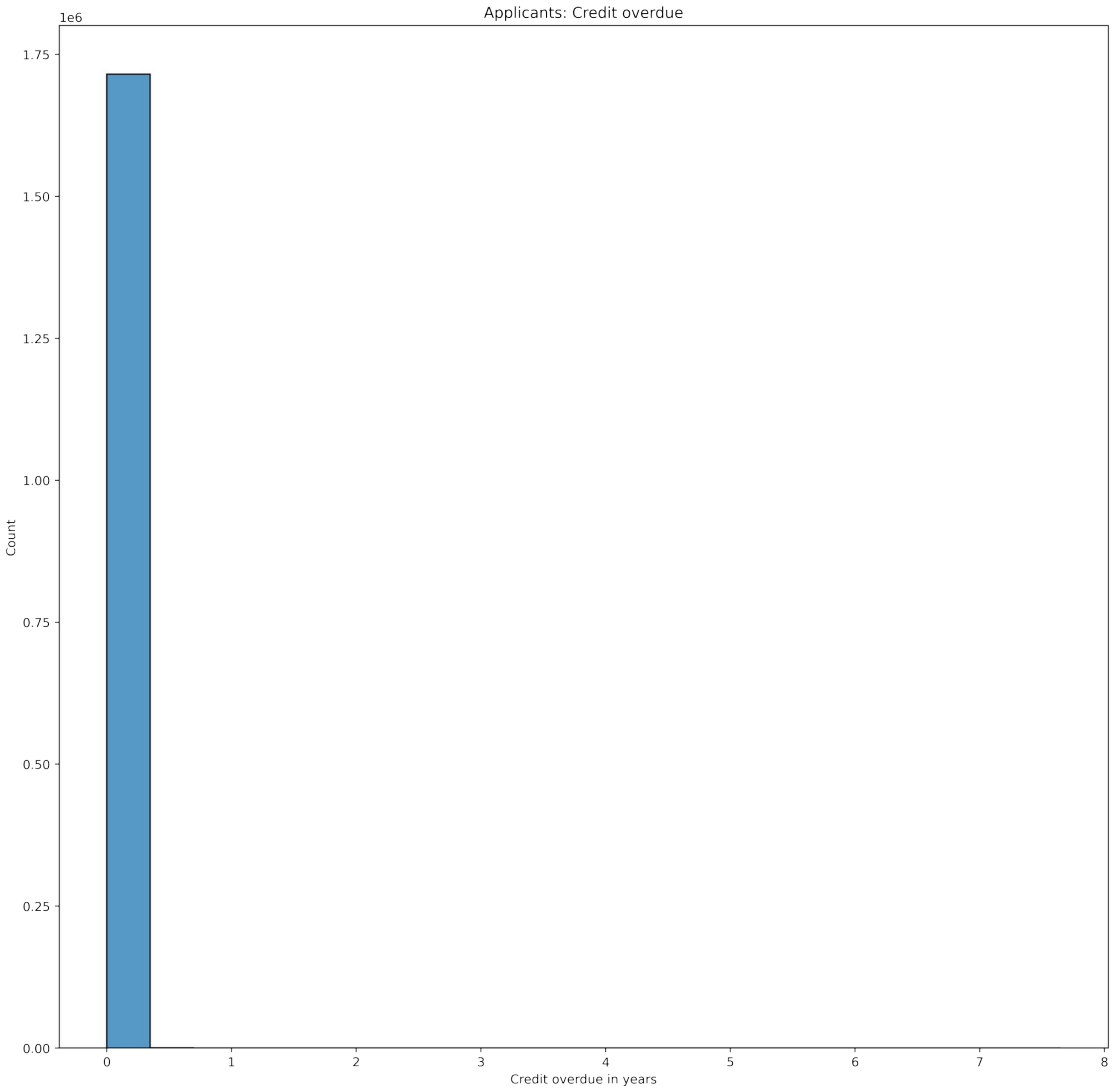
```
fig, ax =  plt.subplots(1,1, figsize=(15,15), dpi=400)
sns.countplot(x="CREDIT_ACTIVE", data=bur,ax=ax)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fae234f8dd0>
```




```
fig, ax =  plt.subplots(1,1, figsize=(15,15), dpi=400)
sns.histplot(bur["CREDIT_DAY_OVERDUE"]/365,ax=ax)
ax.set_title("Applicants: Credit overdue")
ax.set_xlabel("Credit overdue in years")
```

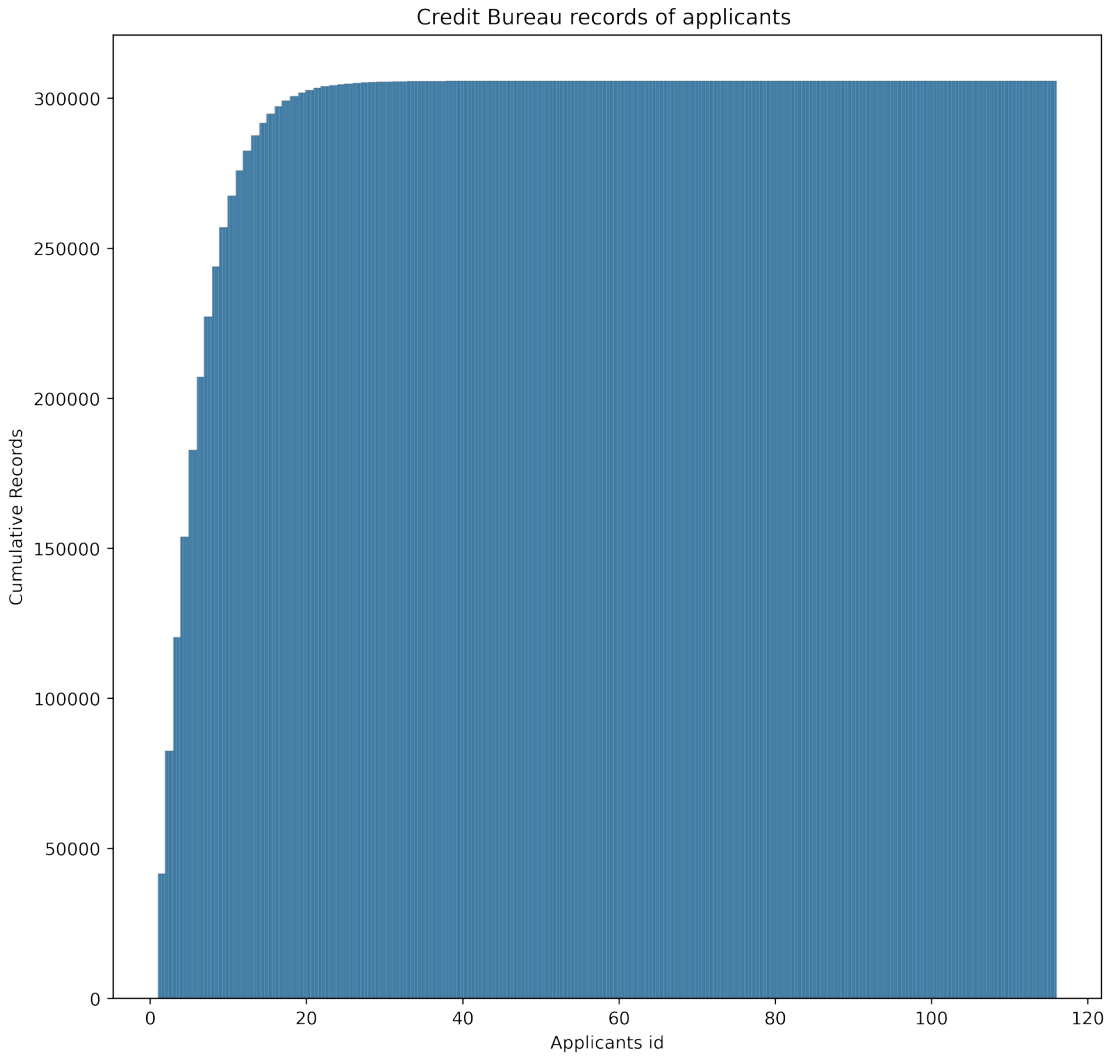
```
Text(0.5, 0, 'Credit overdue in years')
```



Observation: CREDIT_CURRENCY has 4 types but the data majorly contain only one type

```
fig, ax = plt.subplots(1,1, figsize=(10,10), dpi=400)
sns.histplot(bur["SK_ID_CURR"].value_counts().sort_values(),
cumulative=True, ax=ax)
ax.set_title("Credit Bureau records of applicants ")
ax.set_xlabel("Applicants id")
ax.set_ylabel("Cumulative Records")
```

```
Text(0, 0.5, 'Cumulative Records')
```




```
bur_sk_ids = bur["SK_ID_CURR"].value_counts().sort_values()
```

Applicants with more than 5, 10, >15 Bureau records

```
print(" >5 : --> {}\n >10 : --> {}\n >15 : --> {}\n >20 : -->
{}".format(
    len(bur_sk_ids[bur_sk_ids >5]),
    len(bur_sk_ids[bur_sk_ids >10]),
    len(bur_sk_ids[bur_sk_ids >15]),
    len(bur_sk_ids[bur_sk_ids >20]),
))
```

```
 >5 : --> 123079
 >10 : --> 38328
 >15 : --> 10974
 >20 : --> 3079
```

Insights on aggregated data

```
agg_data =
bur.groupby("SK_ID_CURR").agg(['mean','count','sum','min','max'])
agg_data.columns
```

```
MultiIndex([(          'SK_ID_BUREAU',  'mean'),
            (          'SK_ID_BUREAU', 'count'),
            (          'SK_ID_BUREAU',   'sum'),
            (          'SK_ID_BUREAU',   'min'),
            (          'SK_ID_BUREAU',   'max'),
            (           'DAYS_CREDIT',  'mean'),
            (           'DAYS_CREDIT', 'count'),
            (           'DAYS_CREDIT',   'sum'),
            (           'DAYS_CREDIT',   'min'),
            (           'DAYS_CREDIT',   'max'),
            (    'CREDIT_DAY_OVERDUE',  'mean'),
            (    'CREDIT_DAY_OVERDUE', 'count'),
            (    'CREDIT_DAY_OVERDUE',   'sum'),
            (    'CREDIT_DAY_OVERDUE',   'min'),
            (    'CREDIT_DAY_OVERDUE',   'max'),
            (   'DAYS_CREDIT_ENDDATE',  'mean'),
            (   'DAYS_CREDIT_ENDDATE', 'count'),
            (   'DAYS_CREDIT_ENDDATE',   'sum'),
            (   'DAYS_CREDIT_ENDDATE',   'min'),
            (   'DAYS_CREDIT_ENDDATE',   'max'),
            (     'DAYS_ENDDATE_FACT',  'mean'),
            (     'DAYS_ENDDATE_FACT', 'count'),
            (     'DAYS_ENDDATE_FACT',   'sum'),
            (     'DAYS_ENDDATE_FACT',   'min'),
            (     'DAYS_ENDDATE_FACT',   'max'),
            ('AMT_CREDIT_MAX_OVERDUE',  'mean'),
            ('AMT_CREDIT_MAX_OVERDUE', 'count'),
            ('AMT_CREDIT_MAX_OVERDUE',   'sum'),
            ('AMT_CREDIT_MAX_OVERDUE',   'min'),
            ('AMT_CREDIT_MAX_OVERDUE',   'max'),
            (    'CNT_CREDIT_PROLONG',  'mean'),
            (    'CNT_CREDIT_PROLONG', 'count'),
            (    'CNT_CREDIT_PROLONG',   'sum'),
            (    'CNT_CREDIT_PROLONG',   'min'),
            (    'CNT_CREDIT_PROLONG',   'max'),
            (        'AMT_CREDIT_SUM',  'mean'),
            (        'AMT_CREDIT_SUM', 'count'),
            (        'AMT_CREDIT_SUM',   'sum'),
            (        'AMT_CREDIT_SUM',   'min'),
            (        'AMT_CREDIT_SUM',   'max'),
            (   'AMT_CREDIT_SUM_DEBT',  'mean'),
            (   'AMT_CREDIT_SUM_DEBT', 'count'),
            (   'AMT_CREDIT_SUM_DEBT',   'sum'),
            (   'AMT_CREDIT_SUM_DEBT',   'min'),
            (   'AMT_CREDIT_SUM_DEBT',   'max'),
```

```
            (  'AMT_CREDIT_SUM_LIMIT',  'mean'),
            (  'AMT_CREDIT_SUM_LIMIT', 'count'),
            (  'AMT_CREDIT_SUM_LIMIT',   'sum'),
            (  'AMT_CREDIT_SUM_LIMIT',   'min'),
            (  'AMT_CREDIT_SUM_LIMIT',   'max'),
            ('AMT_CREDIT_SUM_OVERDUE',  'mean'),
            ('AMT_CREDIT_SUM_OVERDUE', 'count'),
            ('AMT_CREDIT_SUM_OVERDUE',   'sum'),
            ('AMT_CREDIT_SUM_OVERDUE',   'min'),
            ('AMT_CREDIT_SUM_OVERDUE',   'max'),
            (    'DAYS_CREDIT_UPDATE',  'mean'),
            (    'DAYS_CREDIT_UPDATE', 'count'),
            (    'DAYS_CREDIT_UPDATE',   'sum'),
            (    'DAYS_CREDIT_UPDATE',   'min'),
            (    'DAYS_CREDIT_UPDATE',   'max'),
            (           'AMT_ANNUITY',  'mean'),
            (           'AMT_ANNUITY', 'count'),
            (           'AMT_ANNUITY',   'sum'),
            (           'AMT_ANNUITY',   'min'),
            (           'AMT_ANNUITY',   'max')],
          )

print("-------------DAYS_CREDIT--------------------")
display(agg_data.head(5)["DAYS_CREDIT"])
print("-------------CREDIT_DAY_OVERDUE--------------------")
display(agg_data.head(5)["CREDIT_DAY_OVERDUE"])
print("-------------AMT_CREDIT_MAX_OVERDUE--------------------")
display(agg_data.head(5)["AMT_CREDIT_MAX_OVERDUE"])
print("-------------AMT_CREDIT_SUM--------------------")
display(agg_data.head(5)["AMT_CREDIT_SUM"])
print("-------------AMT_CREDIT_SUM_LIMIT--------------------")
display(agg_data.head(5)["AMT_CREDIT_SUM_LIMIT"])
print("-------------AMT_CREDIT_SUM_OVERDUE--------------------")
display(agg_data.head(5)["AMT_CREDIT_SUM_OVERDUE"])
print("-------------AMT_ANNUITY--------------------")
display(agg_data.head(5)["AMT_ANNUITY"])

-------------DAYS_CREDIT--------------------

                  mean  count   sum   min  max
SK_ID_CURR
100001      -735.000000      7 -5145 -1572  -49
100002      -874.000000      8 -6992 -1437 -103
100003     -1400.750000      4 -5603 -2586 -606
100004      -867.000000      2 -1734 -1326 -408
100005      -190.666667      3  -572  -373  -62

-------------CREDIT_DAY_OVERDUE--------------------

            mean  count  sum  min  max
SK_ID_CURR
```

```
100001       0.0      7    0    0    0
100002       0.0      8    0    0    0
100003       0.0      4    0    0    0
100004       0.0      2    0    0    0
100005       0.0      3    0    0    0

-------------AMT_CREDIT_MAX_OVERDUE--------------------

               mean  count       sum  min       max
SK_ID_CURR
100001          NaN      0     0.000  NaN       NaN
100002      1681.029      5  8405.145  0.0  5043.645
100003         0.000      4     0.000  0.0     0.000
100004         0.000      1     0.000  0.0     0.000
100005         0.000      1     0.000  0.0     0.000

-------------AMT_CREDIT_SUM--------------------

                     mean  count          sum      min       max
SK_ID_CURR
100001      207623.571429      7  1453365.000  85500.0  378000.0
100002      108131.945625      8   865055.565      0.0  450000.0
100003      254350.125000      4  1017400.500  22248.0  810000.0
100004       94518.900000      2   189037.800  94500.0   94537.8
100005      219042.000000      3   657126.000  29826.0  568800.0

-------------AMT_CREDIT_SUM_LIMIT--------------------

                    mean  count         sum  min         max
SK_ID_CURR
100001           0.00000      6       0.000  0.0       0.000
100002         7997.14125      4   31988.565  0.0   31988.565
100003      202500.00000      4  810000.000  0.0  810000.000
100004           0.00000      2       0.000  0.0       0.000
100005           0.00000      3       0.000  0.0       0.000

-------------AMT_CREDIT_SUM_OVERDUE--------------------

            mean  count  sum  min  max
SK_ID_CURR
100001       0.0      7  0.0  0.0  0.0
100002       0.0      8  0.0  0.0  0.0
100003       0.0      4  0.0  0.0  0.0
100004       0.0      2  0.0  0.0  0.0
100005       0.0      3  0.0  0.0  0.0

-------------AMT_ANNUITY--------------------

                   mean  count      sum  min      max
SK_ID_CURR
100001      3545.357143      7  24817.5  0.0  10822.5
100002         0.000000      7      0.0  0.0      0.0
```

```
100003              NaN      0      0.0  NaN      NaN
100004              NaN      0      0.0  NaN      NaN
100005      1420.500000      3   4261.5  0.0   4261.5
```

Dataset: bureau_balance

```
bb = datasets["bureau_balance"]
```

```
bb.head(30)
```

```
    SK_ID_BUREAU  MONTHS_BALANCE STATUS
0        5715448               0      C
1        5715448              -1      C
2        5715448              -2      C
3        5715448              -3      C
4        5715448              -4      C
5        5715448              -5      C
6        5715448              -6      C
7        5715448              -7      C
8        5715448              -8      C
9        5715448              -9      0
10       5715448             -10      0
11       5715448             -11      X
12       5715448             -12      X
13       5715448             -13      X
14       5715448             -14      0
15       5715448             -15      0
16       5715448             -16      0
17       5715448             -17      0
18       5715448             -18      0
19       5715448             -19      0
20       5715448             -20      X
21       5715448             -21      X
22       5715448             -22      X
23       5715448             -23      X
24       5715448             -24      X
25       5715448             -25      X
26       5715448             -26      X
27       5715449               0      C
28       5715449              -1      C
29       5715449              -2      C
```

```
bb.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27299925 entries, 0 to 27299924
Data columns (total 3 columns):
 #   Column          Dtype
---  ------          -----
 0   SK_ID_BUREAU    int64
 1   MONTHS_BALANCE  int64
```

```
 2   STATUS          object
dtypes: int64(2), object(1)
memory usage: 624.8+ MB
```

```
bb.describe()
```

```
       SK_ID_BUREAU  MONTHS_BALANCE
count  2.729992e+07    2.729992e+07
mean   6.036297e+06   -3.074169e+01
std    4.923489e+05    2.386451e+01
min    5.001709e+06   -9.600000e+01
25%    5.730933e+06   -4.600000e+01
50%    6.070821e+06   -2.500000e+01
75%    6.431951e+06   -1.100000e+01
max    6.842888e+06    0.000000e+00
```

```
bb.isna().sum()
```

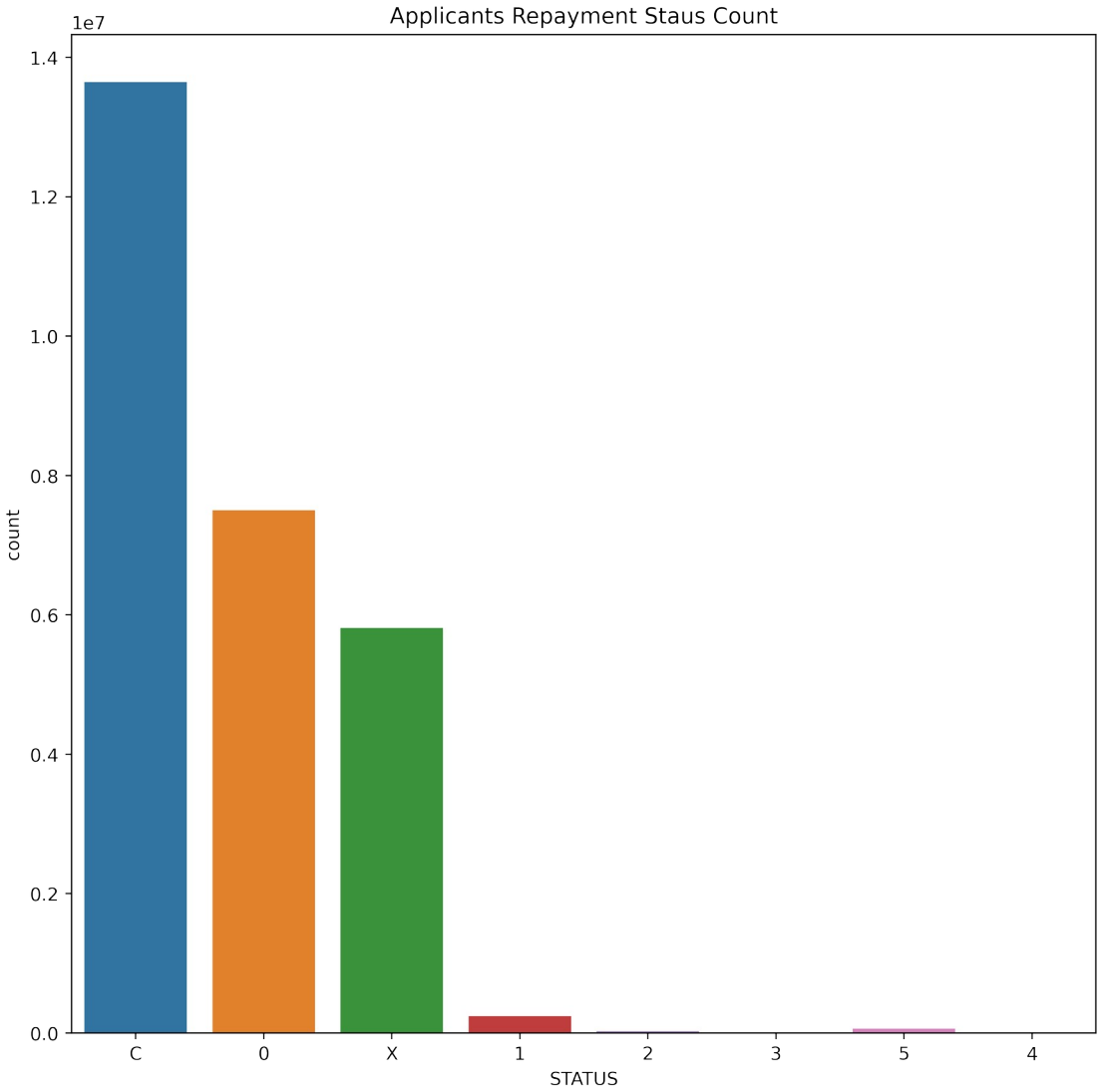
```
SK_ID_BUREAU      0
MONTHS_BALANCE    0
STATUS            0
dtype: int64
```

```
bb.STATUS.unique()
```

```
array(['C', '0', 'X', '1', '2', '3', '5', '4'], dtype=object)
```

```
fig, ax =  plt.subplots(1,1, figsize=(10,10), dpi=400)
sns.countplot(x='STATUS', data=bb,ax=ax)
ax.set_title("Applicants Repayment Staus Count")
```

```
Text(0.5, 1.0, 'Applicants Repayment Staus Count')
```




```
bb.groupby("SK_ID_BUREAU").count().sort_values(by="MONTHS_BALANCE")
```

SK_ID_BUREAU	MONTHS_BALANCE	STATUS
6052856	1	1
5061807	1	1
5918080	1	1
5061817	1	1
6688436	1	1
...	...	...
5359551	97	97
6168534	97	97
6168536	97	97
6395449	97	97
5001709	97	97

[817395 rows x 2 columns]

Observation: we can use the status column to better understand the applicants re-payments behaviour per credit

Questions: What different `status` code indicates, do they have any siginificane??

Dataset: previous_application

```
pa = datasets["previous_application"]
```

```
pa.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1670214 entries, 0 to 1670213
Data columns (total 37 columns):
 #   Column                       Non-Null Count    Dtype  
---  ------                       --------------    -----  
 0   SK_ID_PREV                   1670214 non-null  int64  
 1   SK_ID_CURR                   1670214 non-null  int64  
 2   NAME_CONTRACT_TYPE           1670214 non-null  object 
 3   AMT_ANNUITY                  1297979 non-null  float64
 4   AMT_APPLICATION              1670214 non-null  float64
 5   AMT_CREDIT                   1670213 non-null  float64
 6   AMT_DOWN_PAYMENT             774370 non-null   float64
 7   AMT_GOODS_PRICE              1284699 non-null  float64
 8   WEEKDAY_APPR_PROCESS_START   1670214 non-null  object 
 9   HOUR_APPR_PROCESS_START      1670214 non-null  int64  
 10  FLAG_LAST_APPL_PER_CONTRACT  1670214 non-null  object 
 11  NFLAG_LAST_APPL_IN_DAY       1670214 non-null  int64  
 12  RATE_DOWN_PAYMENT            774370 non-null   float64
 13  RATE_INTEREST_PRIMARY        5951 non-null     float64
 14  RATE_INTEREST_PRIVILEGED     5951 non-null     float64
 15  NAME_CASH_LOAN_PURPOSE       1670214 non-null  object 
 16  NAME_CONTRACT_STATUS         1670214 non-null  object 
 17  DAYS_DECISION                1670214 non-null  int64  
 18  NAME_PAYMENT_TYPE            1670214 non-null  object 
 19  CODE_REJECT_REASON           1670214 non-null  object 
 20  NAME_TYPE_SUITE              849809 non-null   object 
 21  NAME_CLIENT_TYPE             1670214 non-null  object 
 22  NAME_GOODS_CATEGORY          1670214 non-null  object 
 23  NAME_PORTFOLIO               1670214 non-null  object 
 24  NAME_PRODUCT_TYPE            1670214 non-null  object 
 25  CHANNEL_TYPE                 1670214 non-null  object 
 26  SELLERPLACE_AREA             1670214 non-null  int64  
 27  NAME_SELLER_INDUSTRY         1670214 non-null  object 
 28  CNT_PAYMENT                  1297984 non-null  float64
 29  NAME_YIELD_GROUP             1670214 non-null  object 
 30  PRODUCT_COMBINATION          1669868 non-null  object 
 31  DAYS_FIRST_DRAWING           997149 non-null   float64
```

```
 32  DAYS_FIRST_DUE               997149 non-null   float64
 33  DAYS_LAST_DUE_1ST_VERSION    997149 non-null   float64
 34  DAYS_LAST_DUE                997149 non-null   float64
 35  DAYS_TERMINATION             997149 non-null   float64
 36  NFLAG_INSURED_ON_APPROVAL    997149 non-null   float64
dtypes: float64(15), int64(6), object(16)
memory usage: 471.5+ MB

pa.describe()

         SK_ID_PREV    SK_ID_CURR   AMT_ANNUITY  AMT_APPLICATION  \
count  1.670214e+06  1.670214e+06  1.297979e+06     1.670214e+06
mean   1.923089e+06  2.783572e+05  1.595512e+04     1.752339e+05
std    5.325980e+05  1.028148e+05  1.478214e+04     2.927798e+05
min    1.000001e+06  1.000010e+05  0.000000e+00     0.000000e+00
25%    1.461857e+06  1.893290e+05  6.321780e+03     1.872000e+04
50%    1.923110e+06  2.787145e+05  1.125000e+04     7.104600e+04
75%    2.384280e+06  3.675140e+05  2.065842e+04     1.803600e+05
max    2.845382e+06  4.562550e+05  4.180581e+05     6.905160e+06

         AMT_CREDIT  AMT_DOWN_PAYMENT  AMT_GOODS_PRICE  \
count  1.670213e+06      7.743700e+05     1.284699e+06
mean   1.961140e+05      6.697402e+03     2.278473e+05
std    3.185746e+05      2.092150e+04     3.153966e+05
min    0.000000e+00     -9.000000e-01     0.000000e+00
25%    2.416050e+04      0.000000e+00     5.084100e+04
50%    8.054100e+04      1.638000e+03     1.123200e+05
75%    2.164185e+05      7.740000e+03     2.340000e+05
max    6.905160e+06      3.060045e+06     6.905160e+06

       HOUR_APPR_PROCESS_START  NFLAG_LAST_APPL_IN_DAY
RATE_DOWN_PAYMENT  \
count             1.670214e+06            1.670214e+06
774370.000000
mean              1.248418e+01            9.964675e-01
0.079637
std               3.334028e+00            5.932963e-02
0.107823
min               0.000000e+00            0.000000e+00          -
0.000015
25%               1.000000e+01            1.000000e+00
0.000000
50%               1.200000e+01            1.000000e+00
0.051605
75%               1.500000e+01            1.000000e+00
0.108909
max               2.300000e+01            1.000000e+00
1.000000

       ...  RATE_INTEREST_PRIVILEGED  DAYS_DECISION  SELLERPLACE_AREA
```

```
\
count  ...               5951.000000   1.670214e+06      1.670214e+06

mean   ...                  0.773503  -8.806797e+02      3.139511e+02

std    ...                  0.100879   7.790997e+02      7.127443e+03

min    ...                  0.373150  -2.922000e+03     -1.000000e+00

25%    ...                  0.715645  -1.300000e+03     -1.000000e+00

50%    ...                  0.835095  -5.810000e+02      3.000000e+00

75%    ...                  0.852537  -2.800000e+02      8.200000e+01

max    ...                  1.000000  -1.000000e+00      4.000000e+06


        CNT_PAYMENT  DAYS_FIRST_DRAWING  DAYS_FIRST_DUE  \
count  1.297984e+06       997149.000000   997149.000000
mean   1.605408e+01       342209.855039    13826.269337
std    1.456729e+01        88916.115834    72444.869708
min    0.000000e+00        -2922.000000    -2892.000000
25%    6.000000e+00       365243.000000    -1628.000000
50%    1.200000e+01       365243.000000     -831.000000
75%    2.400000e+01       365243.000000     -411.000000
max    8.400000e+01       365243.000000   365243.000000

       DAYS_LAST_DUE_1ST_VERSION  DAYS_LAST_DUE  DAYS_TERMINATION  \
count                 997149.000000  997149.000000     997149.000000
mean                   33767.774054   76582.403064      81992.343838
std                   106857.034789  149647.415123     153303.516729
min                    -2801.000000   -2889.000000      -2874.000000
25%                    -1242.000000   -1314.000000      -1270.000000
50%                     -361.000000    -537.000000       -499.000000
75%                      129.000000     -74.000000        -44.000000
max                   365243.000000  365243.000000     365243.000000

       NFLAG_INSURED_ON_APPROVAL
count                 997149.000000
mean                       0.332570
std                        0.471134
min                        0.000000
25%                        0.000000
50%                        0.000000
75%                        1.000000
max                        1.000000

[8 rows x 21 columns]
```

```
pa.head(10)
```

```
   SK_ID_PREV  SK_ID_CURR NAME_CONTRACT_TYPE  AMT_ANNUITY
AMT_APPLICATION  \
0     2030495      271877     Consumer loans     1730.430
17145.0
1     2802425      108129         Cash loans    25188.615
607500.0
2     2523466      122040         Cash loans    15060.735
112500.0
3     2819243      176158         Cash loans    47041.335
450000.0
4     1784265      202054         Cash loans    31924.395
337500.0
5     1383531      199383         Cash loans    23703.930
315000.0
6     2315218      175704         Cash loans          NaN
0.0
7     1656711      296299         Cash loans          NaN
0.0
8     2367563      342292         Cash loans          NaN
0.0
9     2579447      334349         Cash loans          NaN
0.0

   AMT_CREDIT  AMT_DOWN_PAYMENT  AMT_GOODS_PRICE
WEEKDAY_APPR_PROCESS_START  \
0     17145.0               0.0          17145.0
SATURDAY
1    679671.0               NaN         607500.0
THURSDAY
2    136444.5               NaN         112500.0
TUESDAY
3    470790.0               NaN         450000.0
MONDAY
4    404055.0               NaN         337500.0
THURSDAY
5    340573.5               NaN         315000.0
SATURDAY
6         0.0               NaN              NaN
TUESDAY
7         0.0               NaN              NaN
MONDAY
8         0.0               NaN              NaN
MONDAY
9         0.0               NaN              NaN
SATURDAY

   HOUR_APPR_PROCESS_START  ... NAME_SELLER_INDUSTRY  CNT_PAYMENT  \
0                       15  ...          Connectivity         12.0
```

```
1                       11  ...                  XNA         36.0
2                       11  ...                  XNA         12.0
3                        7  ...                  XNA         12.0
4                        9  ...                  XNA         24.0
5                        8  ...                  XNA         18.0
6                       11  ...                  XNA          NaN
7                        7  ...                  XNA          NaN
8                       15  ...                  XNA          NaN
9                       15  ...                  XNA          NaN

   NAME_YIELD_GROUP       PRODUCT_COMBINATION  DAYS_FIRST_DRAWING  \
0            middle  POS mobile with interest            365243.0
1       low_action          Cash X-Sell: low            365243.0
2              high         Cash X-Sell: high            365243.0
3            middle       Cash X-Sell: middle            365243.0
4              high         Cash Street: high                 NaN
5       low_normal          Cash X-Sell: low            365243.0
6               XNA                      Cash                 NaN
7               XNA                      Cash                 NaN
8               XNA                      Cash                 NaN
9               XNA                      Cash                 NaN

  DAYS_FIRST_DUE DAYS_LAST_DUE_1ST_VERSION  DAYS_LAST_DUE
DAYS_TERMINATION  \
0          -42.0                    300.0          -42.0            -
37.0
1         -134.0                    916.0       365243.0
365243.0
2         -271.0                     59.0       365243.0
365243.0
3         -482.0                   -152.0         -182.0           -
177.0
4            NaN                      NaN            NaN
NaN
5         -654.0                   -144.0         -144.0           -
137.0
6            NaN                      NaN            NaN
NaN
7            NaN                      NaN            NaN
NaN
8            NaN                      NaN            NaN
NaN
9            NaN                      NaN            NaN
NaN

  NFLAG_INSURED_ON_APPROVAL
0                       0.0
1                       1.0
2                       1.0
3                       1.0
```

```
4                       NaN
5                       1.0
6                       NaN
7                       NaN
8                       NaN
9                       NaN

[10 rows x 37 columns]
```

```
pa.isna().sum()
```

```
SK_ID_PREV                         0
SK_ID_CURR                         0
NAME_CONTRACT_TYPE                 0
AMT_ANNUITY                   372235
AMT_APPLICATION                    0
AMT_CREDIT                         1
AMT_DOWN_PAYMENT              895844
AMT_GOODS_PRICE               385515
WEEKDAY_APPR_PROCESS_START         0
HOUR_APPR_PROCESS_START            0
FLAG_LAST_APPL_PER_CONTRACT        0
NFLAG_LAST_APPL_IN_DAY             0
RATE_DOWN_PAYMENT             895844
RATE_INTEREST_PRIMARY        1664263
RATE_INTEREST_PRIVILEGED     1664263
NAME_CASH_LOAN_PURPOSE             0
NAME_CONTRACT_STATUS               0
DAYS_DECISION                      0
NAME_PAYMENT_TYPE                  0
CODE_REJECT_REASON                 0
NAME_TYPE_SUITE               820405
NAME_CLIENT_TYPE                   0
NAME_GOODS_CATEGORY                0
NAME_PORTFOLIO                     0
NAME_PRODUCT_TYPE                  0
CHANNEL_TYPE                       0
SELLERPLACE_AREA                   0
NAME_SELLER_INDUSTRY               0
CNT_PAYMENT                   372230
NAME_YIELD_GROUP                   0
PRODUCT_COMBINATION              346
DAYS_FIRST_DRAWING            673065
DAYS_FIRST_DUE                673065
DAYS_LAST_DUE_1ST_VERSION     673065
DAYS_LAST_DUE                 673065
DAYS_TERMINATION              673065
NFLAG_INSURED_ON_APPROVAL     673065
dtype: int64
```

```
len(pa.SK_ID_CURR.unique())
```

```
338857
```

```
app_train = datasets['application_train']
app_train_req = app_train[["SK_ID_CURR", "TARGET"]]
merged_df = pd.merge(pa, app_train_req, how="left")
```

```
merged_df
```

```
         SK_ID_PREV  SK_ID_CURR NAME_CONTRACT_TYPE  AMT_ANNUITY  \
0           2030495      271877     Consumer loans     1730.430
1           2802425      108129         Cash loans    25188.615
2           2523466      122040         Cash loans    15060.735
3           2819243      176158         Cash loans    47041.335
4           1784265      202054         Cash loans    31924.395
...             ...         ...                ...          ...
1670209     2300464      352015     Consumer loans    14704.290
1670210     2357031      334635     Consumer loans     6622.020
1670211     2659632      249544     Consumer loans    11520.855
1670212     2785582      400317         Cash loans    18821.520
1670213     2418762      261212         Cash loans    16431.300

         AMT_APPLICATION  AMT_CREDIT  AMT_DOWN_PAYMENT
AMT_GOODS_PRICE  \
0                17145.0     17145.0              0.0
17145.0
1               607500.0    679671.0              NaN
607500.0
2               112500.0    136444.5              NaN
112500.0
3               450000.0    470790.0              NaN
450000.0
4               337500.0    404055.0              NaN
337500.0
...                  ...         ...              ...              ..
.
1670209         267295.5    311400.0              0.0
267295.5
1670210          87750.0     64291.5          29250.0
87750.0
1670211         105237.0    102523.5          10525.5
105237.0
1670212         180000.0    191880.0              NaN
180000.0
1670213         360000.0    360000.0              NaN
360000.0

        WEEKDAY_APPR_PROCESS_START  HOUR_APPR_PROCESS_START  ...
CNT_PAYMENT  \
0                         SATURDAY                       15  ...
12.0
1                         THURSDAY                       11  ...
```

```
36.0
2                         TUESDAY                         11  ...
12.0
3                          MONDAY                          7  ...
12.0
4                        THURSDAY                          9  ...
24.0
...                           ...                        ...  ...
...
1670209                  WEDNESDAY                         12  ...
30.0
1670210                    TUESDAY                         15  ...
12.0
1670211                     MONDAY                         12  ...
10.0
1670212                  WEDNESDAY                          9  ...
12.0
1670213                     SUNDAY                         10  ...
48.0

         NAME_YIELD_GROUP          PRODUCT_COMBINATION
DAYS_FIRST_DRAWING  \
0                  middle     POS mobile with interest
365243.0
1             low_action             Cash X-Sell: low
365243.0
2                   high            Cash X-Sell: high
365243.0
3                  middle          Cash X-Sell: middle
365243.0
4                   high            Cash Street: high
NaN
...                  ...                           ...
...
1670209        low_normal   POS industry with interest
365243.0
1670210            middle   POS industry with interest
365243.0
1670211        low_normal  POS household with interest
365243.0
1670212        low_normal             Cash X-Sell: low
365243.0
1670213            middle          Cash X-Sell: middle
365243.0

         DAYS_FIRST_DUE DAYS_LAST_DUE_1ST_VERSION DAYS_LAST_DUE  \
0                 -42.0                     300.0         -42.0
1                -134.0                     916.0      365243.0
2                -271.0                      59.0      365243.0
3                -482.0                    -152.0        -182.0
```

```
4                   NaN                         NaN           NaN
...                 ...                         ...           ...
1670209          -508.0                       362.0        -358.0
1670210         -1604.0                     -1274.0       -1304.0
1670211         -1457.0                     -1187.0       -1187.0
1670212         -1155.0                      -825.0        -825.0
1670213         -1163.0                       247.0        -443.0

         DAYS_TERMINATION NFLAG_INSURED_ON_APPROVAL TARGET
0                   -37.0                       0.0    0.0
1                365243.0                       1.0    0.0
2                365243.0                       1.0    0.0
3                  -177.0                       1.0    0.0
4                     NaN                       NaN    0.0
...                   ...                       ...    ...
1670209            -351.0                       0.0    NaN
1670210           -1297.0                       0.0    NaN
1670211           -1181.0                       0.0    0.0
1670212            -817.0                       1.0    0.0
1670213            -423.0                       0.0    0.0

[1670214 rows x 38 columns]
```

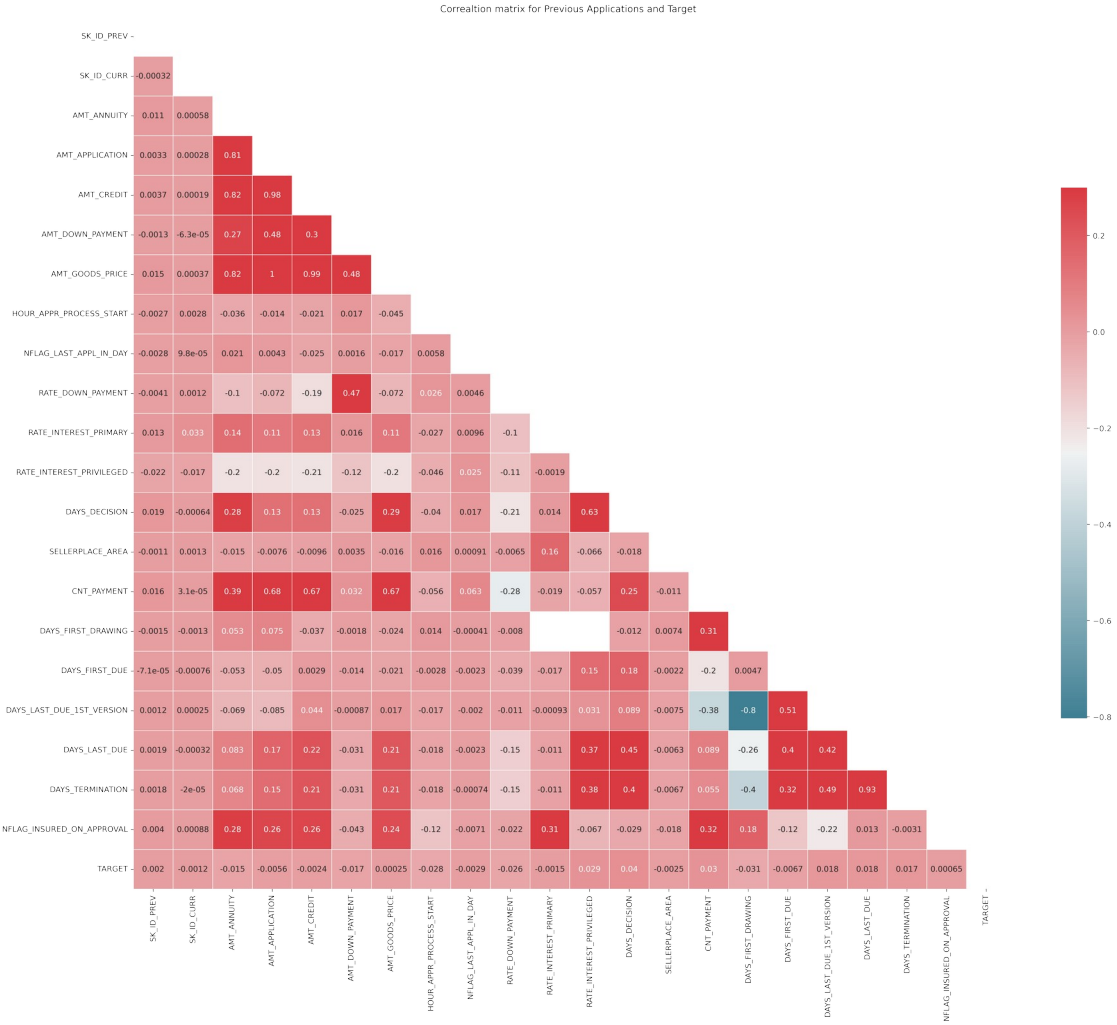
```
merged_df.corr()["TARGET"].sort_values()
```

```
DAYS_FIRST_DRAWING          -0.031154
HOUR_APPR_PROCESS_START     -0.027809
RATE_DOWN_PAYMENT           -0.026111
AMT_DOWN_PAYMENT            -0.016918
AMT_ANNUITY                 -0.014922
DAYS_FIRST_DUE              -0.006651
AMT_APPLICATION             -0.005583
NFLAG_LAST_APPL_IN_DAY      -0.002887
SELLERPLACE_AREA            -0.002539
AMT_CREDIT                  -0.002350
RATE_INTEREST_PRIMARY       -0.001470
SK_ID_CURR                  -0.001246
AMT_GOODS_PRICE              0.000254
NFLAG_INSURED_ON_APPROVAL    0.000653
SK_ID_PREV                   0.002009
DAYS_TERMINATION             0.016981
DAYS_LAST_DUE                0.017522
DAYS_LAST_DUE_1ST_VERSION    0.018021
RATE_INTEREST_PRIVILEGED     0.028640
CNT_PAYMENT                  0.030480
DAYS_DECISION                0.039901
TARGET                       1.000000
Name: TARGET, dtype: float64
```

```
# set up the matplotlib figure
f, ax = plt.subplots(1,1, figsize=(25, 25), dpi=400)
```

```
# generate a mask for the lower triangle
mask = np.zeros_like(merged_df.corr(), dtype=np.bool_)
mask[np.triu_indices_from(mask)] = True

# generate a custom diverging colormap
cmap = sns.diverging_palette(220, 11, as_cmap=True)

# draw the heatmap with the mask and correct aspect ratio
sns.heatmap(merged_df.corr(), mask=mask, cmap=cmap, vmax=.3,
            square=True,
            linewidths=.5, cbar_kws={"shrink": .5}, ax=ax,
annot=True);
ax.set_title("Correaltion matrix for Previous Applications and Target
")
```

```
Text(0.5, 1.0, 'Correaltion matrix for Previous Applications and
Target ')
```



```
pa.groupby("SK_ID_CURR").count()
```

```
            SK_ID_PREV  NAME_CONTRACT_TYPE  AMT_ANNUITY
AMT_APPLICATION  \
SK_ID_CURR

100001               1                   1            1
1
100002               1                   1            1
1
100003               3                   3            3
3
100004               1                   1            1
1
100005               2                   2            1
2
...                ...                 ...          ...              .
..
456251               1                   1            1
1
456252               1                   1            1
1
456253               2                   2            2
2
456254               2                   2            2
2
456255               8                   8            8
8

            AMT_CREDIT  AMT_DOWN_PAYMENT  AMT_GOODS_PRICE  \
SK_ID_CURR
100001               1                 1                1
100002               1                 1                1
100003               3                 2                3
100004               1                 1                1
100005               2                 1                1
...                ...               ...              ...
456251               1                 1                1
456252               1                 1                1
456253               2                 2                2
456254               2                 2                2
456255               8                 3                8

            WEEKDAY_APPR_PROCESS_START  HOUR_APPR_PROCESS_START  \
SK_ID_CURR
100001                               1                       1
100002                               1                       1
100003                               3                       3
100004                               1                       1
100005                               2                       2
```

```
...                                  ...                       ...
456251                                 1                         1
456252                                 1                         1
456253                                 2                         2
456254                                 2                         2
456255                                 8                         8

            FLAG_LAST_APPL_PER_CONTRACT  ...  NAME_SELLER_INDUSTRY  \
SK_ID_CURR                                ...
100001                                1  ...                     1
100002                                1  ...                     1
100003                                3  ...                     3
100004                                1  ...                     1
100005                                2  ...                     2
...                                 ...  ...                   ...
456251                                1  ...                     1
456252                                1  ...                     1
456253                                2  ...                     2
456254                                2  ...                     2
456255                                8  ...                     8

            CNT_PAYMENT  NAME_YIELD_GROUP  PRODUCT_COMBINATION  \
SK_ID_CURR
100001                1                1                    1
100002                1                1                    1
100003                3                3                    3
100004                1                1                    1
100005                1                2                    2
...                 ...              ...                  ...
456251                1                1                    1
456252                1                1                    1
456253                2                2                    2
456254                2                2                    2
456255                8                8                    8

            DAYS_FIRST_DRAWING  DAYS_FIRST_DUE
DAYS_LAST_DUE_1ST_VERSION  \
SK_ID_CURR

100001                       1              1
1
100002                       1              1
1
100003                       3              3
3
100004                       1              1
1
100005                       1              1
1
...                        ...            ...
```

```
...
456251                        1               1
1
456252                        1               1
1
456253                        2               2
2
456254                        2               2
2
456255                        6               6
6
```

SK_ID_CURR	DAYS_LAST_DUE	DAYS_TERMINATION	NFLAG_INSURED_ON_APPROVAL
100001	1	1	1
100002	1	1	1
100003	3	3	3
100004	1	1	1
100005	1	1	1
...	...	...	...
456251	1	1	1
456252	1	1	1
456253	2	2	2
456254	2	2	2
456255	6	6	6

[338857 rows x 36 columns]

Features can be used from `previous_application` when grouped by "SK_ID_CURR"

```
app_train_req[app_train_req.TARGET==1].head(20)
```

	SK_ID_CURR	TARGET
0	100002	1
26	100031	1
40	100047	1

```
42       100049       1
81       100096       1
94       100112       1
110      100130       1
138      100160       1
154      100181       1
163      100192       1
180      100209       1
184      100214       1
211      100246       1
235      100273       1
242      100282       1
246      100286       1
255      100295       1
260      100300       1
261      100301       1
283      100326       1
```

```
observation_ids = [100003, 100192, 100286]
for grp, df in pa.groupby("SK_ID_CURR"):
    if grp in observation_ids:
        display(pd.merge(df, app_train_req, how="left"))
        observation_ids.remove(grp)
        if len(observation_ids) ==0:
            break
if len(observation_ids) !=0:
    print("NO PREVIOUS APPLICATION FOUND FOR :
{}".format(observation_ids))
```

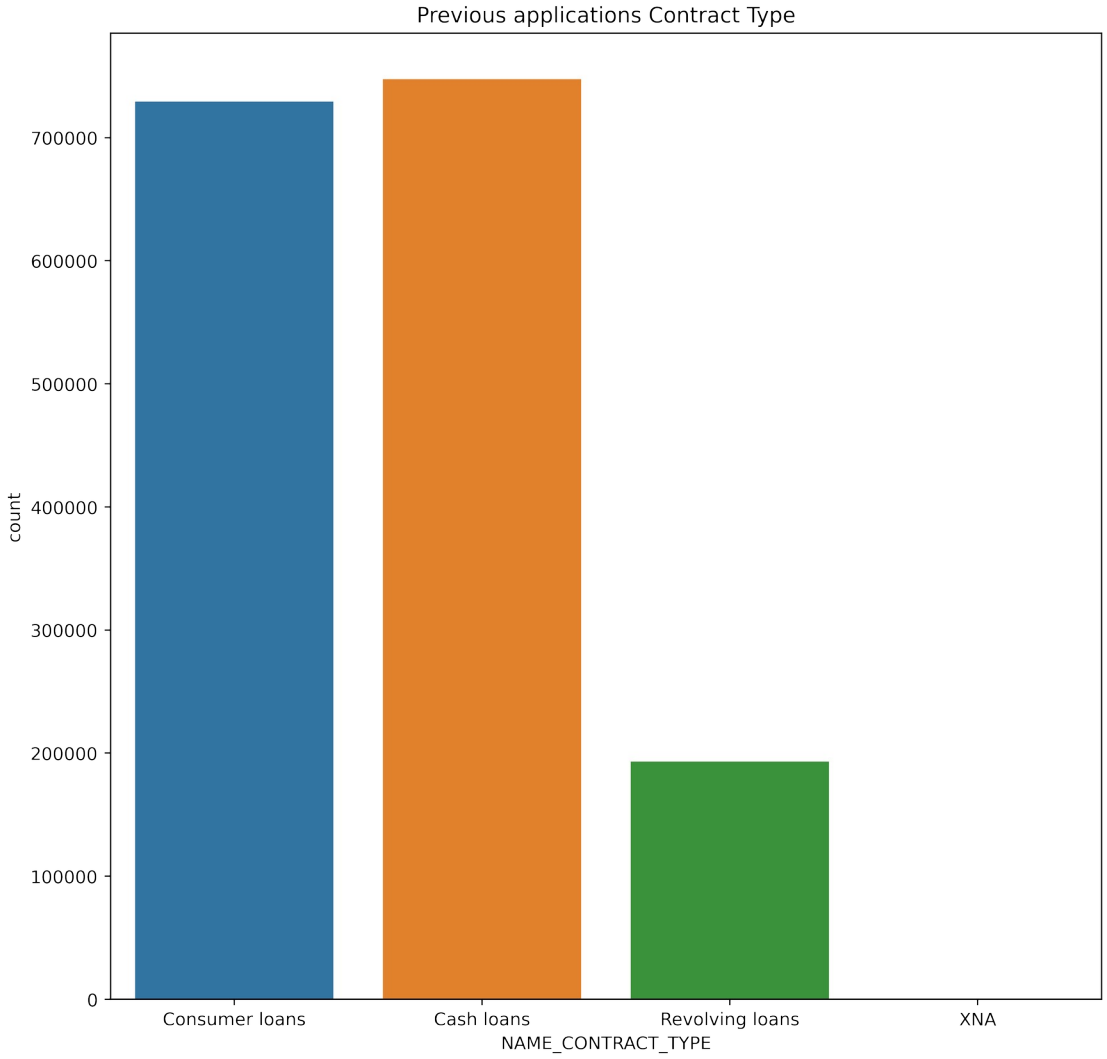
```
   SK_ID_PREV  SK_ID_CURR NAME_CONTRACT_TYPE  AMT_ANNUITY
AMT_APPLICATION  \
0     1810518      100003         Cash loans    98356.995
900000.0
1     2636178      100003     Consumer loans    64567.665
337500.0
2     2396755      100003     Consumer loans     6737.310
68809.5

   AMT_CREDIT  AMT_DOWN_PAYMENT  AMT_GOODS_PRICE
WEEKDAY_APPR_PROCESS_START  \
0   1035882.0               NaN         900000.0
FRIDAY
1    348637.5               0.0         337500.0
SUNDAY
2     68053.5            6885.0          68809.5
SATURDAY

   HOUR_APPR_PROCESS_START  ... CNT_PAYMENT  NAME_YIELD_GROUP  \
0                       12  ...        12.0        low_normal
1                       17  ...         6.0            middle
```

```
2                      15  ...        12.0            middle

           PRODUCT_COMBINATION  DAYS_FIRST_DRAWING  DAYS_FIRST_DUE  \
0             Cash X-Sell: low            365243.0          -716.0
1   POS industry with interest            365243.0          -797.0
2  POS household with interest            365243.0         -2310.0

  DAYS_LAST_DUE_1ST_VERSION DAYS_LAST_DUE  DAYS_TERMINATION  \
0                    -386.0        -536.0            -527.0
1                    -647.0        -647.0            -639.0
2                   -1980.0       -1980.0           -1976.0

  NFLAG_INSURED_ON_APPROVAL TARGET
0                       1.0      0
1                       0.0      0
2                       1.0      0

[3 rows x 38 columns]

   SK_ID_PREV  SK_ID_CURR NAME_CONTRACT_TYPE  AMT_ANNUITY
AMT_APPLICATION  \
0     1185166      100286     Consumer loans      4889.97
47385.0

   AMT_CREDIT  AMT_DOWN_PAYMENT  AMT_GOODS_PRICE
WEEKDAY_APPR_PROCESS_START  \
0     52389.0               0.0          47385.0
SATURDAY

   HOUR_APPR_PROCESS_START  ... CNT_PAYMENT  NAME_YIELD_GROUP  \
0                        8  ...        12.0        low_normal

           PRODUCT_COMBINATION  DAYS_FIRST_DRAWING  DAYS_FIRST_DUE  \
0  POS household with interest            365243.0          -207.0

  DAYS_LAST_DUE_1ST_VERSION DAYS_LAST_DUE  DAYS_TERMINATION  \
0                     123.0      365243.0          365243.0

  NFLAG_INSURED_ON_APPROVAL TARGET
0                       0.0      1

[1 rows x 38 columns]

NO PREVIOUS APPLICATION FOUND FOR :[100192]
```

Design Question: How to navigate around situation where no previous_application exits for SK_ID_CURR

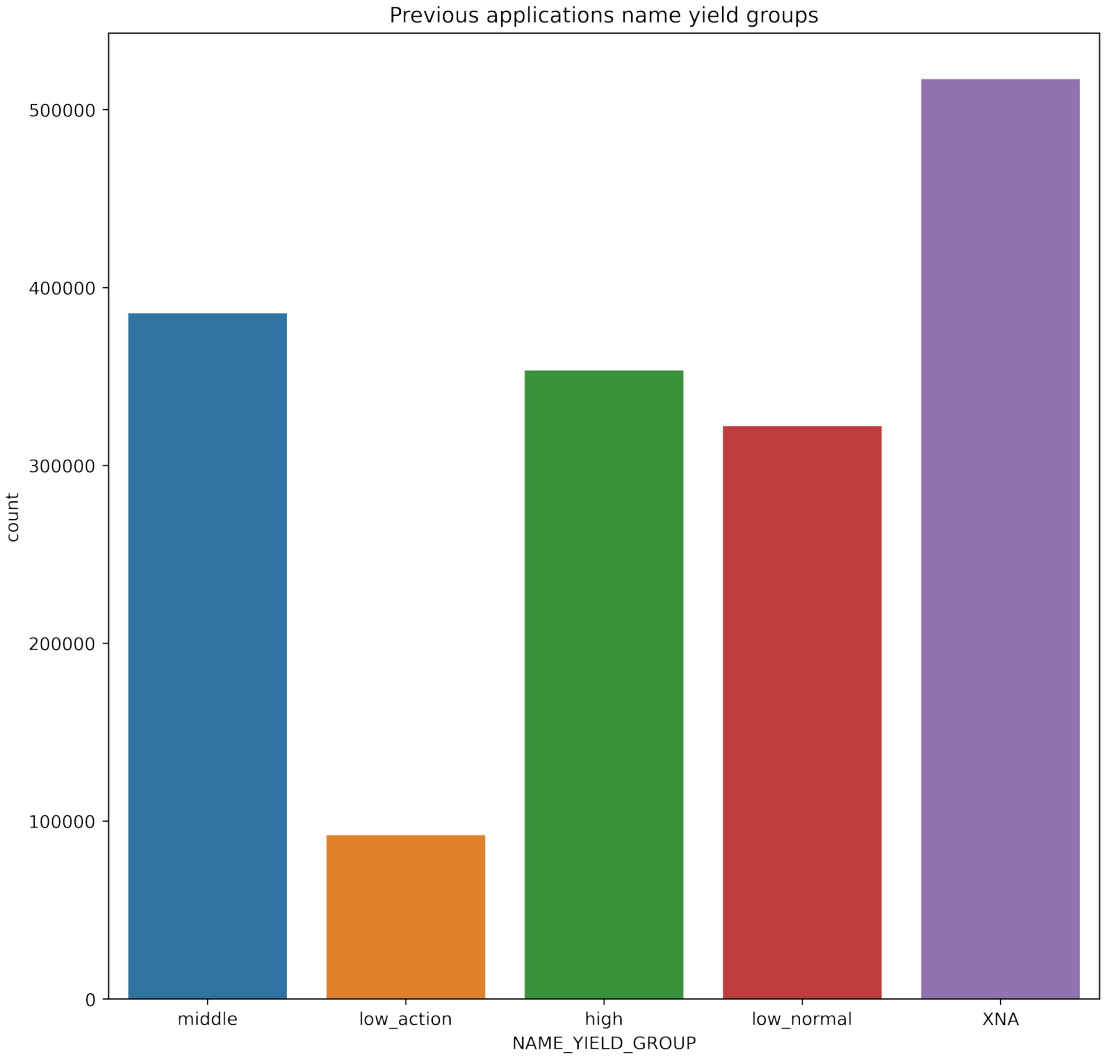
```
fig, ax =  plt.subplots(1,1, figsize=(10,10), dpi=400)
sns.countplot(x='NAME_CONTRACT_TYPE', data=pa,ax=ax)
ax.set_title("Previous applications Contract Type")
```

```
Text(0.5, 1.0, 'Previous applications Contract Type')
```




```
fig, ax =  plt.subplots(1,1, figsize=(10,10), dpi=400)
sns.countplot(x='NAME_YIELD_GROUP', data=pa,ax=ax)
ax.set_title("Previous applications name yield groups")
```

```
Text(0.5, 1.0, 'Previous applications name yield groups')
```

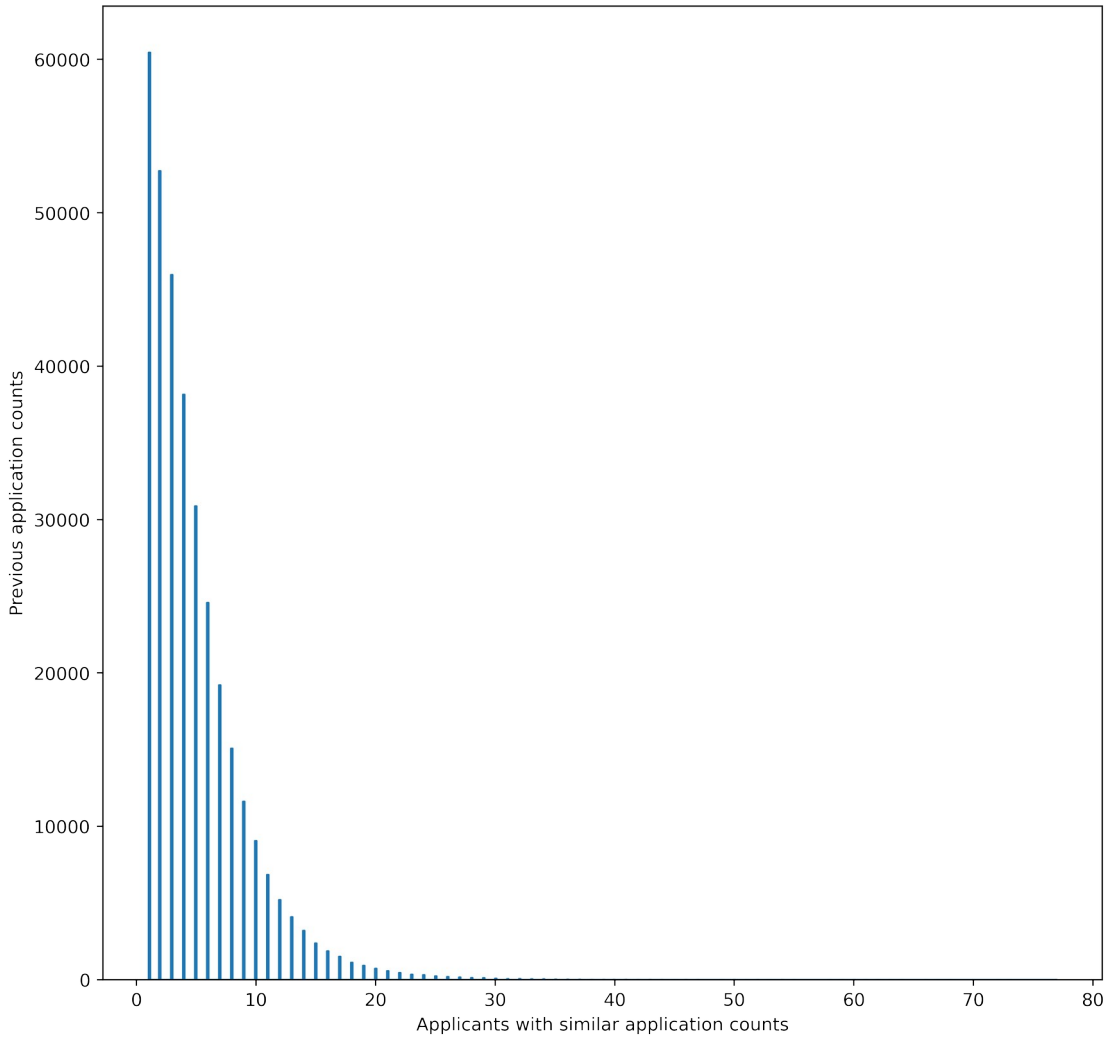



```
pa_sk_ids = pa['SK_ID_CURR'].value_counts()

len(pa_sk_ids[pa_sk_ids==1])
```

```
60458
```

```
fig, ax =  plt.subplots(1,1, figsize=(10,10), dpi=400)

sns.histplot(pa['SK_ID_CURR'].value_counts(), element="step", ax=ax)
ax.set_ylabel("Previous application counts")
ax.set_xlabel("Applicants with similar application counts")
```

```
Text(0.5, 0, 'Applicants with similar application counts')
```



Dataset: `credit_card_balance`

```
ccb = datasets["credit_card_balance"]
```

```
ccb.head(10)
```

```
   SK_ID_PREV  SK_ID_CURR  MONTHS_BALANCE  AMT_BALANCE  \
0     2562384      378907              -6       56.970
1     2582071      363914              -1    63975.555
2     1740877      371185              -7    31815.225
3     1389973      337855              -4   236572.110
4     1891521      126868              -1   453919.455
5     2646502      380010              -7    82903.815
6     1079071      171320              -6   353451.645
7     2095912      118650              -7    47962.125
8     2181852      367360              -4   291543.075
9     1235299      203885              -5   201261.195
```

```
   AMT_CREDIT_LIMIT_ACTUAL  AMT_DRAWINGS_ATM_CURRENT
AMT_DRAWINGS_CURRENT  \
0                   135000                      0.0
877.500
1                    45000                   2250.0
2250.000
2                   450000                      0.0
0.000
3                   225000                   2250.0
2250.000
4                   450000                      0.0
11547.000
5                   270000                      0.0
0.000
6                   585000                  67500.0
67500.000
7                    45000                  45000.0
45000.000
8                   292500                  90000.0
289339.425
9                   225000                  76500.0
111026.700

   AMT_DRAWINGS_OTHER_CURRENT  AMT_DRAWINGS_POS_CURRENT  \
0                         0.0                   877.500
1                         0.0                     0.000
2                         0.0                     0.000
3                         0.0                     0.000
4                         0.0                 11547.000
5                         0.0                     0.000
6                         0.0                     0.000
7                         0.0                     0.000
8                         0.0                199339.425
9                         0.0                 34526.700

   AMT_INST_MIN_REGULARITY  ...  AMT_RECIVABLE
AMT_TOTAL_RECEIVABLE  \
0                 1700.325  ...          0.000                0.000

1                 2250.000  ...      64875.555            64875.555

2                 2250.000  ...      31460.085            31460.085

3                11795.760  ...     233048.970           233048.970

4                22924.890  ...     453919.455           453919.455

5                 4449.105  ...      82773.315            82773.315
```

```
6                14684.175  ...     351881.145            351881.145

7                    0.000  ...      47962.125             47962.125

8                  130.500  ...     286831.575            286831.575

9                 6338.340  ...     197224.695            197224.695


   CNT_DRAWINGS_ATM_CURRENT  CNT_DRAWINGS_CURRENT
CNT_DRAWINGS_OTHER_CURRENT  \
0                       0.0                     1
0.0
1                       1.0                     1
0.0
2                       0.0                     0
0.0
3                       1.0                     1
0.0
4                       0.0                     1
0.0
5                       0.0                     0
0.0
6                       1.0                     1
0.0
7                       1.0                     1
0.0
8                       3.0                     8
0.0
9                       3.0                     9
0.0

   CNT_DRAWINGS_POS_CURRENT  CNT_INSTALMENT_MATURE_CUM
NAME_CONTRACT_STATUS  \
0                       1.0                      35.0
Active
1                       0.0                      69.0
Active
2                       0.0                      30.0
Active
3                       0.0                      10.0
Active
4                       1.0                     101.0
Active
5                       0.0                       2.0
Active
6                       0.0                       6.0
Active
7                       0.0                      51.0
Active
```

```
8                       5.0                        3.0
Active
9                       6.0                       38.0
Active

   SK_DPD  SK_DPD_DEF
0       0           0
1       0           0
2       0           0
3       0           0
4       0           0
5       7           0
6       0           0
7       0           0
8       0           0
9       0           0

[10 rows x 23 columns]
```

```
ccb.describe()
```

```
         SK_ID_PREV    SK_ID_CURR  MONTHS_BALANCE   AMT_BALANCE  \
count  3.840312e+06  3.840312e+06    3.840312e+06  3.840312e+06
mean   1.904504e+06  2.783242e+05   -3.452192e+01  5.830016e+04
std    5.364695e+05  1.027045e+05    2.666775e+01  1.063070e+05
min    1.000018e+06  1.000060e+05   -9.600000e+01 -4.202502e+05
25%    1.434385e+06  1.895170e+05   -5.500000e+01  0.000000e+00
50%    1.897122e+06  2.783960e+05   -2.800000e+01  0.000000e+00
75%    2.369328e+06  3.675800e+05   -1.100000e+01  8.904669e+04
max    2.843496e+06  4.562500e+05   -1.000000e+00  1.505902e+06

       AMT_CREDIT_LIMIT_ACTUAL  AMT_DRAWINGS_ATM_CURRENT  \
count             3.840312e+06              3.090496e+06
mean              1.538080e+05              5.961325e+03
std               1.651457e+05              2.822569e+04
min               0.000000e+00             -6.827310e+03
25%               4.500000e+04              0.000000e+00
50%               1.125000e+05              0.000000e+00
75%               1.800000e+05              0.000000e+00
max               1.350000e+06              2.115000e+06

       AMT_DRAWINGS_CURRENT  AMT_DRAWINGS_OTHER_CURRENT  \
count          3.840312e+06                3.090496e+06
mean           7.433388e+03                2.881696e+02
std            3.384608e+04                8.201989e+03
min           -6.211620e+03                0.000000e+00
25%            0.000000e+00                0.000000e+00
50%            0.000000e+00                0.000000e+00
75%            0.000000e+00                0.000000e+00
max            2.287098e+06                1.529847e+06
```

```
       AMT_DRAWINGS_POS_CURRENT  AMT_INST_MIN_REGULARITY  ...  \
count              3.090496e+06             3.535076e+06  ...
mean               2.968805e+03             3.540204e+03  ...
std                2.079689e+04             5.600154e+03  ...
min                0.000000e+00             0.000000e+00  ...
25%                0.000000e+00             0.000000e+00  ...
50%                0.000000e+00             0.000000e+00  ...
75%                0.000000e+00             6.633911e+03  ...
max                2.239274e+06             2.028820e+05  ...

       AMT_RECEIVABLE_PRINCIPAL  AMT_RECIVABLE
AMT_TOTAL_RECEIVABLE  \
count              3.840312e+06   3.840312e+06          3.840312e+06

mean               5.596588e+04   5.808881e+04          5.809829e+04

std                1.025336e+05   1.059654e+05          1.059718e+05

min               -4.233058e+05  -4.202502e+05         -4.202502e+05

25%                0.000000e+00   0.000000e+00          0.000000e+00

50%                0.000000e+00   0.000000e+00          0.000000e+00

75%                8.535924e+04   8.889949e+04          8.891451e+04

max                1.472317e+06   1.493338e+06          1.493338e+06


       CNT_DRAWINGS_ATM_CURRENT  CNT_DRAWINGS_CURRENT  \
count              3.090496e+06          3.840312e+06
mean               3.094490e-01          7.031439e-01
std                1.100401e+00          3.190347e+00
min                0.000000e+00          0.000000e+00
25%                0.000000e+00          0.000000e+00
50%                0.000000e+00          0.000000e+00
75%                0.000000e+00          0.000000e+00
max                5.100000e+01          1.650000e+02

       CNT_DRAWINGS_OTHER_CURRENT  CNT_DRAWINGS_POS_CURRENT  \
count                3.090496e+06              3.090496e+06
mean                 4.812496e-03              5.594791e-01
std                  8.263861e-02              3.240649e+00
min                  0.000000e+00              0.000000e+00
25%                  0.000000e+00              0.000000e+00
50%                  0.000000e+00              0.000000e+00
75%                  0.000000e+00              0.000000e+00
max                  1.200000e+01              1.650000e+02
```

```
       CNT_INSTALMENT_MATURE_CUM        SK_DPD    SK_DPD_DEF
count               3.535076e+06  3.840312e+06  3.840312e+06
mean                2.082508e+01  9.283667e+00  3.316220e-01
std                 2.005149e+01  9.751570e+01  2.147923e+01
min                 0.000000e+00  0.000000e+00  0.000000e+00
25%                 4.000000e+00  0.000000e+00  0.000000e+00
50%                 1.500000e+01  0.000000e+00  0.000000e+00
75%                 3.200000e+01  0.000000e+00  0.000000e+00
max                 1.200000e+02  3.260000e+03  3.260000e+03

[8 rows x 22 columns]
```

```
ccb.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3840312 entries, 0 to 3840311
Data columns (total 23 columns):
 #   Column                        Dtype
---  ------                        -----
 0   SK_ID_PREV                    int64
 1   SK_ID_CURR                    int64
 2   MONTHS_BALANCE                int64
 3   AMT_BALANCE                   float64
 4   AMT_CREDIT_LIMIT_ACTUAL       int64
 5   AMT_DRAWINGS_ATM_CURRENT      float64
 6   AMT_DRAWINGS_CURRENT          float64
 7   AMT_DRAWINGS_OTHER_CURRENT    float64
 8   AMT_DRAWINGS_POS_CURRENT      float64
 9   AMT_INST_MIN_REGULARITY       float64
 10  AMT_PAYMENT_CURRENT           float64
 11  AMT_PAYMENT_TOTAL_CURRENT     float64
 12  AMT_RECEIVABLE_PRINCIPAL      float64
 13  AMT_RECIVABLE                 float64
 14  AMT_TOTAL_RECEIVABLE          float64
 15  CNT_DRAWINGS_ATM_CURRENT      float64
 16  CNT_DRAWINGS_CURRENT          int64
 17  CNT_DRAWINGS_OTHER_CURRENT    float64
 18  CNT_DRAWINGS_POS_CURRENT      float64
 19  CNT_INSTALMENT_MATURE_CUM     float64
 20  NAME_CONTRACT_STATUS          object
 21  SK_DPD                        int64
 22  SK_DPD_DEF                    int64
dtypes: float64(15), int64(7), object(1)
memory usage: 673.9+ MB
```

```
ccb.isna().sum()
```

```
SK_ID_PREV                      0
SK_ID_CURR                      0
MONTHS_BALANCE                  0
```

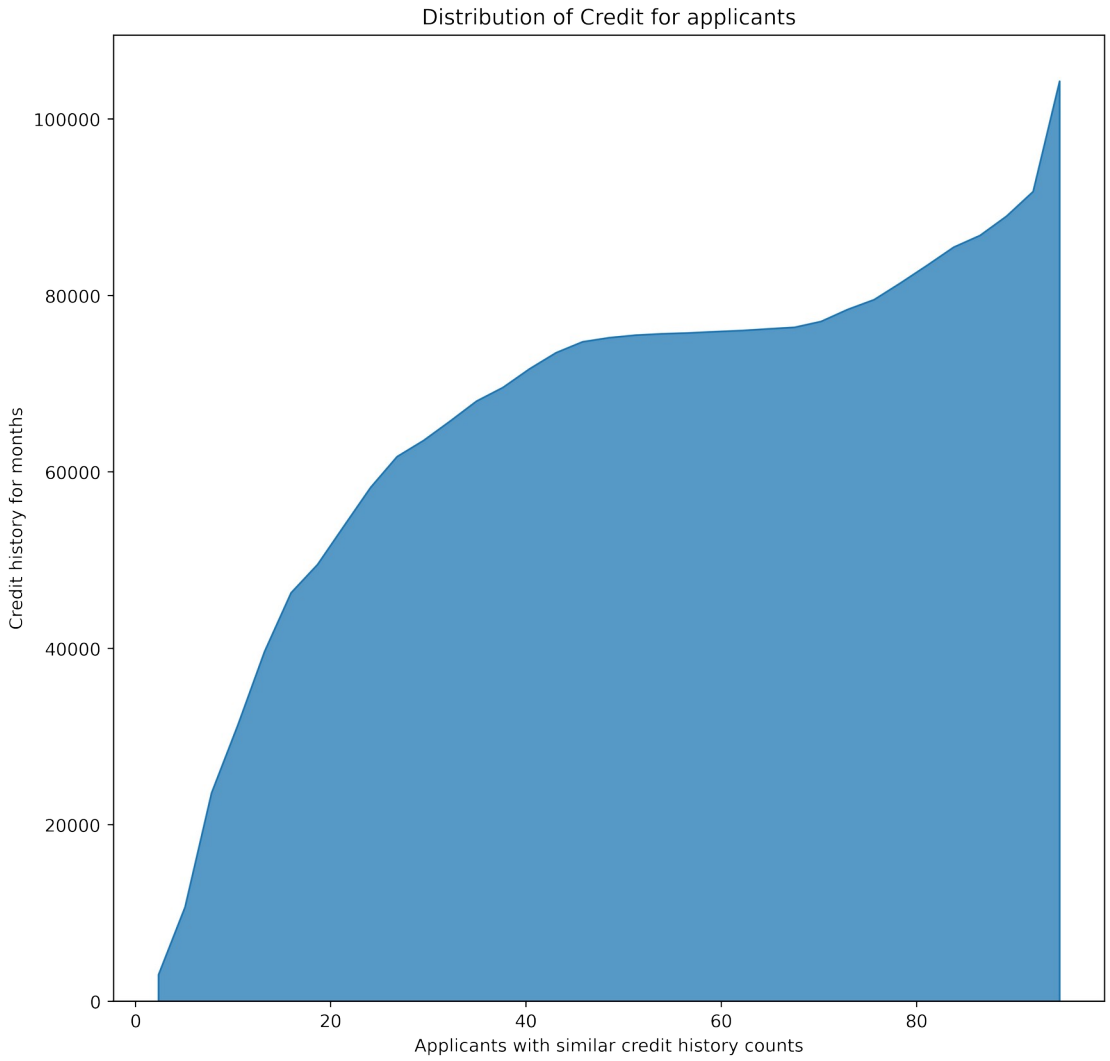
```
AMT_BALANCE                         0
AMT_CREDIT_LIMIT_ACTUAL             0
AMT_DRAWINGS_ATM_CURRENT       749816
AMT_DRAWINGS_CURRENT                0
AMT_DRAWINGS_OTHER_CURRENT     749816
AMT_DRAWINGS_POS_CURRENT       749816
AMT_INST_MIN_REGULARITY        305236
AMT_PAYMENT_CURRENT            767988
AMT_PAYMENT_TOTAL_CURRENT           0
AMT_RECEIVABLE_PRINCIPAL            0
AMT_RECIVABLE                       0
AMT_TOTAL_RECEIVABLE                0
CNT_DRAWINGS_ATM_CURRENT       749816
CNT_DRAWINGS_CURRENT                0
CNT_DRAWINGS_OTHER_CURRENT     749816
CNT_DRAWINGS_POS_CURRENT       749816
CNT_INSTALMENT_MATURE_CUM      305236
NAME_CONTRACT_STATUS                0
SK_DPD                              0
SK_DPD_DEF                          0
dtype: int64
```

```
ccb["SK_ID_PREV"].value_counts(dropna=False).sort_values()
```

```
2191610     1
1383311     1
2697963     1
1083843     1
2636347     1
           ..
1025633    96
2045163    96
1839174    96
1824679    96
2377894    96
Name: SK_ID_PREV, Length: 104307, dtype: int64
```

```
fig, ax =  plt.subplots(1,1, figsize=(10,10), dpi=400)

sns.histplot(ccb['SK_ID_PREV'].value_counts(), element="poly", ax=ax,
cumulative=True)
ax.set_ylabel("Credit history for months")
ax.set_xlabel("Applicants with similar credit history counts")

ax.set_title("Distribution of Credit for applicants")
```

```
Text(0.5, 1.0, 'Distribution of Credit for applicants')
```



Selecting random SK_ID_PREV to gain insights

```
app_train = datasets['application_train']
app_train_req = app_train[["SK_ID_CURR", "TARGET"]]

app_train_ccb = pd.merge(app_train_req,
ccb[["SK_ID_CURR","SK_ID_PREV"]], how="left")
app_train_ccb.fillna(0, inplace=True)
app_train_ccb.SK_ID_PREV=app_train_ccb.SK_ID_PREV.astype(int)

display(app_train_ccb[app_train_ccb.TARGET==1].sort_values(by="SK_ID_P
REV",ascending=False).head(5))
display(app_train_ccb[app_train_ccb.TARGET==0].sort_values(by="SK_ID_P
REV",ascending=False).head(5))
```

	SK_ID_CURR	TARGET	SK_ID_PREV
1067881	210644	1	2843461
1067868	210644	1	2843461
1067874	210644	1	2843461

```
1067873      210644       1     2843461
1067872      210644       1     2843461

         SK_ID_CURR  TARGET  SK_ID_PREV
2295551      337804       0     2843493
2295555      337804       0     2843493
2295547      337804       0     2843493
2295548      337804       0     2843493
2295552      337804       0     2843493
```

```
observation_ids = [2843461, 2843493, 2843478]
for grp, df in ccb.groupby("SK_ID_PREV"):
    if grp in observation_ids:
        display(pd.merge(df, app_train_req,
how="left").sort_values(by="MONTHS_BALANCE", ascending=False))
        observation_ids.remove(grp)
        if len(observation_ids) ==0:
            break
```

```
    SK_ID_PREV  SK_ID_CURR  MONTHS_BALANCE  AMT_BALANCE  \
14     2843461      210644              -1    44589.735
5      2843461      210644              -2    45904.095
31     2843461      210644              -3    37647.225
27     2843461      210644              -4    39085.695
12     2843461      210644              -5    39776.490
..         ...         ...             ...          ...
13     2843461      210644             -69    65340.000
46     2843461      210644             -70    66995.280
11     2843461      210644             -71    82152.180
20     2843461      210644             -72    78989.535
68     2843461      210644             -73    89685.000

    AMT_CREDIT_LIMIT_ACTUAL  AMT_DRAWINGS_ATM_CURRENT
AMT_DRAWINGS_CURRENT  \
14                    45000                      0.0
0.0
5                     45000                      0.0
9243.0
31                    45000                      0.0
0.0
27                    45000                      0.0
1165.5
12                    45000                      0.0
0.0
..                      ...                      ...
...
13                    90000                      0.0
0.0
46                    90000                      0.0
0.0
11                    90000                      0.0
```

```
0.0
20                    90000                         0.0
0.0
68                    90000                         0.0
87660.0

    AMT_DRAWINGS_OTHER_CURRENT  AMT_DRAWINGS_POS_CURRENT  \
14                         0.0                       0.0
5                          0.0                    9243.0
31                         0.0                       0.0
27                         0.0                    1165.5
12                         0.0                       0.0
..                         ...                       ...
13                         0.0                       0.0
46                         0.0                       0.0
11                         0.0                       0.0
20                         0.0                       0.0
68                         0.0                   87660.0

    AMT_INST_MIN_REGULARITY  ...  AMT_TOTAL_RECEIVABLE  \
14                   2250.0  ...              44589.735
5                    2250.0  ...              45904.095
31                   2250.0  ...              37647.225
27                   2250.0  ...              39085.695
12                   2250.0  ...              39776.490
..                      ...  ...                    ...
13                   4500.0  ...              65340.000
46                   4500.0  ...              66995.280
11                   4500.0  ...              82152.180
20                   4500.0  ...              78989.535
68                      NaN  ...              89685.000

    CNT_DRAWINGS_ATM_CURRENT  CNT_DRAWINGS_CURRENT  \
14                       0.0                     0
5                        0.0                     1
31                       0.0                     0
27                       0.0                     1
12                       0.0                     0
..                       ...                   ...
13                       0.0                     0
46                       0.0                     0
11                       0.0                     0
20                       0.0                     0
68                       0.0                     2

    CNT_DRAWINGS_OTHER_CURRENT  CNT_DRAWINGS_POS_CURRENT  \
14                         0.0                       0.0
5                          0.0                       1.0
31                         0.0                       0.0
27                         0.0                       1.0
```

```
12                          0.0                         0.0
..                          ...                         ...
13                          0.0                         0.0
46                          0.0                         0.0
11                          0.0                         0.0
20                          0.0                         0.0
68                          0.0                         2.0

    CNT_INSTALMENT_MATURE_CUM  NAME_CONTRACT_STATUS  SK_DPD
SK_DPD_DEF  \
14                       51.0                Active       0
0
5                        50.0                Active       0
0
31                       49.0                Active       0
0
27                       48.0                Active       0
0
12                       47.0                Active       0
0
..                        ...                   ...     ...         ..
.
13                        4.0                Active       0
0
46                        3.0                Active       0
0
11                        2.0                Active       1
1
20                        1.0                Active       0
0
68                        NaN                Active       0
0

    TARGET
14       1
5        1
31       1
27       1
12       1
..     ...
13       1
46       1
11       1
20       1
68       1

[73 rows x 24 columns]

    SK_ID_PREV  SK_ID_CURR  MONTHS_BALANCE  AMT_BALANCE  \
42     2843478      424526              -2        0.000
```

```
2      2843478      424526              -3        0.000
32     2843478      424526              -4        0.000
11     2843478      424526              -5        0.000
45     2843478      424526              -6        0.000
..         ...         ...             ...          ...
26     2843478      424526             -87    59249.475
69     2843478      424526             -88    71238.330
70     2843478      424526             -89    77646.555
47     2843478      424526             -90    66824.820
5      2843478      424526             -91    69369.750

    AMT_CREDIT_LIMIT_ACTUAL  AMT_DRAWINGS_ATM_CURRENT
AMT_DRAWINGS_CURRENT  \
42                        0                      0.0
0.0
2                         0                      0.0
0.0
32                        0                      0.0
0.0
11                        0                      0.0
0.0
45                        0                      0.0
0.0
..                      ...                      ...
...
26                    90000                      0.0
0.0
69                    90000                      0.0
0.0
70                    90000                  22500.0
22500.0
47                    90000                      0.0
0.0
5                     90000                  67500.0
67500.0

    AMT_DRAWINGS_OTHER_CURRENT  AMT_DRAWINGS_POS_CURRENT  \
42                         0.0                       0.0
2                          0.0                       0.0
32                         0.0                       0.0
11                         0.0                       0.0
45                         0.0                       0.0
..                         ...                       ...
26                         0.0                       0.0
69                         0.0                       0.0
70                         0.0                       0.0
47                         0.0                       0.0
5                          0.0                       0.0

    AMT_INST_MIN_REGULARITY  ...  AMT_TOTAL_RECEIVABLE  \
```

```
42                      0.0  ...                 0.000
2                       0.0  ...                 0.000
32                      0.0  ...                 0.000
11                      0.0  ...                 0.000
45                      0.0  ...                 0.000
..                      ...  ...                   ...
26                   4500.0  ...             59249.475
69                   4500.0  ...             71238.330
70                   4500.0  ...             77646.555
47                   4500.0  ...             66824.820
5                       NaN  ...             69369.750

    CNT_DRAWINGS_ATM_CURRENT  CNT_DRAWINGS_CURRENT  \
42                       0.0                     0
2                        0.0                     0
32                       0.0                     0
11                       0.0                     0
45                       0.0                     0
..                       ...                   ...
26                       0.0                     0
69                       0.0                     0
70                       1.0                     1
47                       0.0                     0
5                        3.0                     3

    CNT_DRAWINGS_OTHER_CURRENT  CNT_DRAWINGS_POS_CURRENT  \
42                         0.0                       0.0
2                          0.0                       0.0
32                         0.0                       0.0
11                         0.0                       0.0
45                         0.0                       0.0
..                         ...                       ...
26                         0.0                       0.0
69                         0.0                       0.0
70                         0.0                       0.0
47                         0.0                       0.0
5                          0.0                       0.0

    CNT_INSTALMENT_MATURE_CUM  NAME_CONTRACT_STATUS  SK_DPD
SK_DPD_DEF  \
42                       10.0                Active       0
0
2                        10.0                Active       0
0
32                       10.0                Active       0
0
11                       10.0                Active       0
0
45                       10.0                Active       0
0
```

```
..                        ...                   ...     ...         ..
.
26                        4.0                Active       0
0
69                        3.0                Active       0
0
70                        2.0                Active       0
0
47                        1.0                Active       0
0
5                         NaN                Active       0
0

    TARGET
42       0
2        0
32       0
11       0
45       0
..     ...
26       0
69       0
70       0
47       0
5        0

[90 rows x 24 columns]

    SK_ID_PREV  SK_ID_CURR  MONTHS_BALANCE  AMT_BALANCE  \
14     2843493      337804              -1    70445.790
3      2843493      337804              -2    74304.135
4      2843493      337804              -3    75970.080
2      2843493      337804              -4    72921.015
7      2843493      337804              -5    78839.595
6      2843493      337804              -6    60719.130
1      2843493      337804              -7    62644.140
13     2843493      337804              -8    66804.030
0      2843493      337804              -9    68541.165
8      2843493      337804             -10    70222.185
12     2843493      337804             -11    70159.635
5      2843493      337804             -12    64969.380
10     2843493      337804             -13    25036.515
11     2843493      337804             -14    25522.110
9      2843493      337804             -15        0.000

    AMT_CREDIT_LIMIT_ACTUAL  AMT_DRAWINGS_ATM_CURRENT
AMT_DRAWINGS_CURRENT  \
14                   225000                      0.0
0.0
3                    225000                      0.0
```

```
0.0
4                     225000                         0.0
0.0
2                     225000                         0.0
0.0
7                     225000                         0.0
28998.0
6                     135000                         0.0
0.0
1                     135000                         0.0
0.0
13                    135000                         0.0
0.0
0                     135000                         0.0
0.0
8                     135000                         0.0
0.0
12                    135000                         0.0
3565.8
5                     135000                         0.0
47839.5
10                     90000                         0.0
2969.1
11                     90000                         0.0
24885.0
9                      45000                         NaN
0.0

    AMT_DRAWINGS_OTHER_CURRENT  AMT_DRAWINGS_POS_CURRENT  \
14                         0.0                       0.0
3                          0.0                       0.0
4                          0.0                       0.0
2                          0.0                       0.0
7                          0.0                   28998.0
6                          0.0                       0.0
1                          0.0                       0.0
13                         0.0                       0.0
0                          0.0                       0.0
8                          0.0                       0.0
12                         0.0                    3565.8
5                          0.0                   47839.5
10                         0.0                    2969.1
11                         0.0                   24885.0
9                          NaN                       NaN

    AMT_INST_MIN_REGULARITY  ...  AMT_TOTAL_RECEIVABLE  \
14                  3674.790  ...              69261.345
3                   3748.455  ...              73091.025
4                   4043.655  ...              74746.170
2                   4358.970  ...              72921.015
```

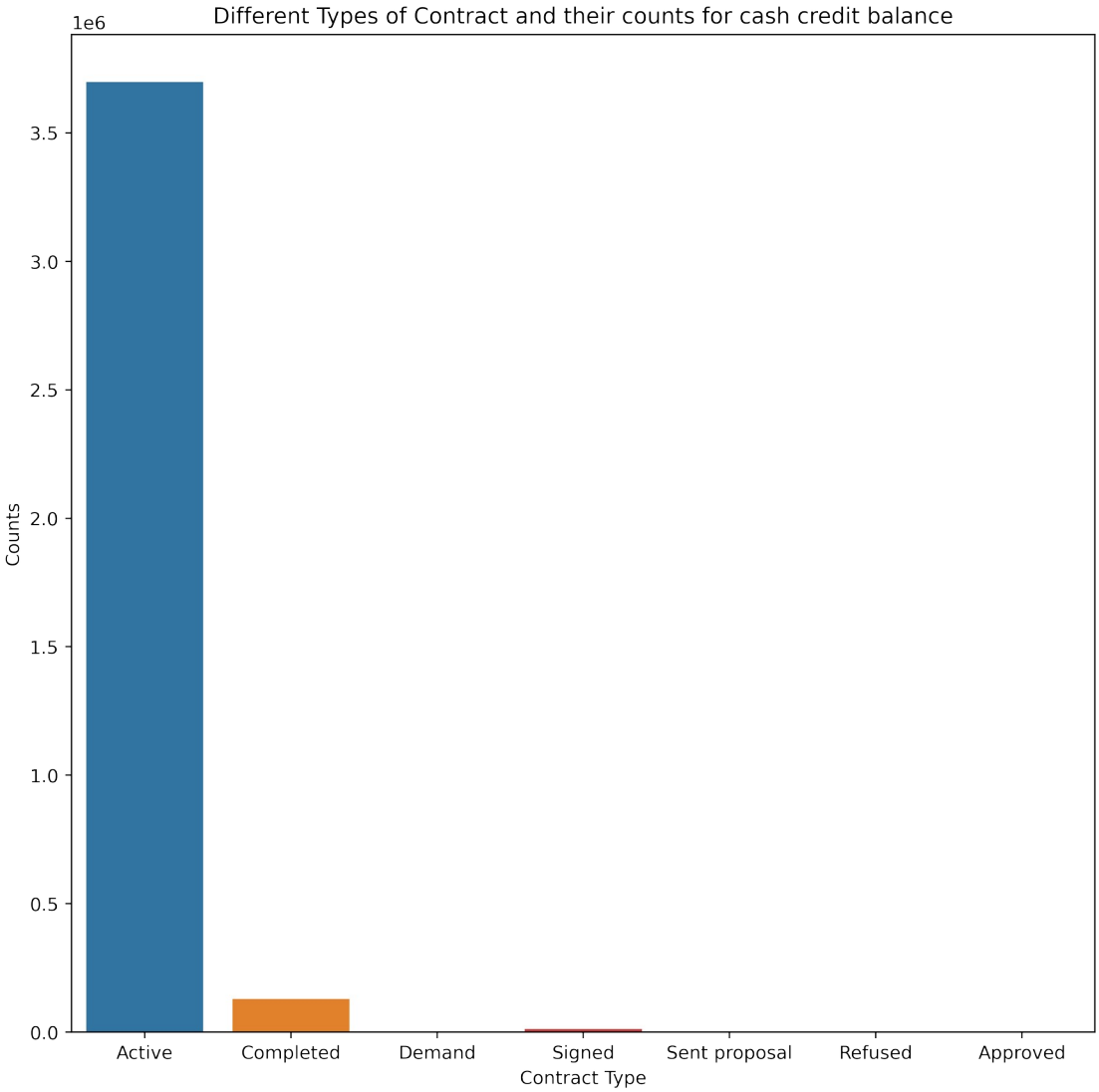
```
7                  2996.010  ...             78839.595
6                  3093.435  ...             59606.910
1                  3296.250  ...             61517.970
13                 3381.885  ...             65647.350
0                  3468.240  ...             67371.210
8                  3496.230  ...             69047.910
12                 2250.000  ...             68963.445
5                  2250.000  ...             64969.380
10                 2250.000  ...             24903.945
11                    0.000  ...             24885.000
9                     0.000  ...                 0.000

    CNT_DRAWINGS_ATM_CURRENT  CNT_DRAWINGS_CURRENT  \
14                       0.0                     0
3                        0.0                     0
4                        0.0                     0
2                        0.0                     0
7                        0.0                     1
6                        0.0                     0
1                        0.0                     0
13                       0.0                     0
0                        0.0                     0
8                        0.0                     0
12                       0.0                     1
5                        0.0                     2
10                       0.0                     1
11                       0.0                     2
9                        NaN                     0

    CNT_DRAWINGS_OTHER_CURRENT  CNT_DRAWINGS_POS_CURRENT  \
14                         0.0                       0.0
3                          0.0                       0.0
4                          0.0                       0.0
2                          0.0                       0.0
7                          0.0                       1.0
6                          0.0                       0.0
1                          0.0                       0.0
13                         0.0                       0.0
0                          0.0                       0.0
8                          0.0                       0.0
12                         0.0                       1.0
5                          0.0                       2.0
10                         0.0                       1.0
11                         0.0                       2.0
9                          NaN                       NaN

    CNT_INSTALMENT_MATURE_CUM  NAME_CONTRACT_STATUS  SK_DPD
SK_DPD_DEF  \
14                       13.0                Active       0
0
```

```
3                          12.0                Active       0
0
4                          11.0                Active       0
0
2                          10.0                Active       0
0
7                           9.0                Active       0
0
6                           8.0                Active       0
0
1                           7.0                Active       0
0
13                          6.0                Active       0
0
0                           5.0                Active       0
0
8                           4.0                Active       0
0
12                          3.0                Active       0
0
5                           2.0                Active       0
0
10                          1.0                Active       0
0
11                          0.0                Active       0
0
9                           0.0                Active       0
0

    TARGET
14       0
3        0
4        0
2        0
7        0
6        0
1        0
13       0
0        0
8        0
12       0
5        0
10       0
11       0
9        0

[15 rows x 24 columns]
```

```
fig, ax =  plt.subplots(1,1, figsize=(10,10), dpi=400)
sns.countplot(x='NAME_CONTRACT_STATUS', data=ccb,ax=ax)
```

```
ax.set_xlabel("Contract Type")
ax.set_ylabel("Counts")
ax.set_title("Different Types of Contract and their counts for cash
credit balance")
```

```
Text(0.5, 1.0, 'Different Types of Contract and their counts for cash
credit balance')
```




```
merged_df = pd.merge(ccb, app_train_req, how="left", on="SK_ID_CURR")

# set up the matplotlib figure
f, ax = plt.subplots(1,1, figsize=(25, 25), dpi=500)

# generate a mask for the lower triangle
mask = np.zeros_like(merged_df.corr(), dtype=np.bool_)
mask[np.triu_indices_from(mask)] = True
```

```
# generate a custom diverging colormap
cmap = sns.diverging_palette(220, 11, as_cmap=True)

# draw the heatmap with the mask and correct aspect ratio
sns.heatmap(merged_df.corr(), mask=mask, cmap=cmap, vmax=.3,
            square=True,
            linewidths=.5, cbar_kws={"shrink": .5}, ax=ax,
annot=True);
ax.set_title("Correaltion matrix for Cash Credit Balance and Target ")
```

```
Output hidden; open in https://colab.research.google.com to view.
```

Insights on aggregated data

```
agg_data =
ccb.groupby("SK_ID_PREV").agg(['mean','count','sum','min','max'])
agg_data.columns
```

```
MultiIndex([(    'SK_ID_CURR',  'mean'),
            (    'SK_ID_CURR', 'count'),
            (    'SK_ID_CURR',   'sum'),
            (    'SK_ID_CURR',   'min'),
            (    'SK_ID_CURR',   'max'),
            ('MONTHS_BALANCE',  'mean'),
            ('MONTHS_BALANCE', 'count'),
            ('MONTHS_BALANCE',   'sum'),
            ('MONTHS_BALANCE',   'min'),
            ('MONTHS_BALANCE',   'max'),
            ...
            (        'SK_DPD',  'mean'),
            (        'SK_DPD', 'count'),
            (        'SK_DPD',   'sum'),
            (        'SK_DPD',   'min'),
            (        'SK_DPD',   'max'),
            (    'SK_DPD_DEF',  'mean'),
            (    'SK_DPD_DEF', 'count'),
            (    'SK_DPD_DEF',   'sum'),
            (    'SK_DPD_DEF',   'min'),
            (    'SK_DPD_DEF',   'max')],
           length=105)
```

```
# dir(agg_data.columns)
agg_data.columns.levels
```

```
FrozenList([['SK_ID_CURR', 'MONTHS_BALANCE', 'AMT_BALANCE',
'AMT_CREDIT_LIMIT_ACTUAL', 'AMT_DRAWINGS_ATM_CURRENT',
'AMT_DRAWINGS_CURRENT', 'AMT_DRAWINGS_OTHER_CURRENT',
'AMT_DRAWINGS_POS_CURRENT', 'AMT_INST_MIN_REGULARITY',
'AMT_PAYMENT_CURRENT', 'AMT_PAYMENT_TOTAL_CURRENT',
'AMT_RECEIVABLE_PRINCIPAL', 'AMT_RECIVABLE', 'AMT_TOTAL_RECEIVABLE',
'CNT_DRAWINGS_ATM_CURRENT', 'CNT_DRAWINGS_CURRENT',
'CNT_DRAWINGS_OTHER_CURRENT', 'CNT_DRAWINGS_POS_CURRENT',
```

```
'CNT_INSTALMENT_MATURE_CUM', 'SK_DPD', 'SK_DPD_DEF'], ['mean',
'count', 'sum', 'min', 'max']])

print("-------------MONTHS_BALANCE--------------------")
display(agg_data.head(5)["MONTHS_BALANCE"])
print("-------------AMT_BALANCE--------------------")
display(agg_data.head(5)["AMT_BALANCE"])
print("-------------AMT_CREDIT_LIMIT_ACTUAL--------------------")
display(agg_data.head(5)["AMT_CREDIT_LIMIT_ACTUAL"])
print("-------------AMT_PAYMENT_CURRENT--------------------")
display(agg_data.head(5)["AMT_PAYMENT_CURRENT"])
print("-------------AMT_RECEIVABLE_PRINCIPAL--------------------")
display(agg_data.head(5)["AMT_RECEIVABLE_PRINCIPAL"])
```

```
-------------MONTHS_BALANCE--------------------
```

SK_ID_PREV	mean	count	sum	min	max
1000018	-4.0	5	-20	-6	-2
1000030	-4.5	8	-36	-8	-1
1000031	-8.5	16	-136	-16	-1
1000035	-4.0	5	-20	-6	-2
1000077	-7.0	11	-77	-12	-2

```
-------------AMT_BALANCE--------------------
```

SK_ID_PREV	mean	count	sum	min	max
1000018	74946.285000	5	374731.425	38879.145	136695.420
1000030	55991.064375	8	447928.515	0.000	103027.275
1000031	52394.439375	16	838311.030	0.000	154945.935
1000035	0.000000	5	0.000	0.000	0.000
1000077	0.000000	11	0.000	0.000	0.000

```
-------------AMT_CREDIT_LIMIT_ACTUAL--------------------
```

SK_ID_PREV	mean	count	sum	min	max
1000018	81000.000000	5	405000	45000	135000
1000030	81562.500000	8	652500	45000	135000
1000031	149625.000000	16	2394000	45000	225000
1000035	225000.000000	5	1125000	225000	225000
1000077	94090.909091	11	1035000	45000	135000

```
-------------AMT_PAYMENT_CURRENT--------------------
```

SK_ID_PREV	mean	count	sum	min	max
1000018	5541.750000	5	27708.75	3190.635	9000.00
1000030	6188.631429	7	43320.42	2371.815	16067.25
1000031	29543.257500	12	354519.09	394.065	160606.80

```
1000035              NaN      0       0.00       NaN        NaN
1000077              NaN      0       0.00       NaN        NaN

-------------AMT_RECEIVABLE_PRINCIPAL--------------------

                    mean  count         sum        min         max
SK_ID_PREV
1000018     72298.197000      5  361490.985  37542.645  132903.000
1000030     55474.453125      8  443795.625      0.000  101866.725
1000031     51402.878437     16  822446.055      0.000  154945.935
1000035         0.000000      5       0.000      0.000       0.000
1000077         0.000000     11       0.000      0.000       0.000
```

Dataset: POS_CASH_BALANCE

```
pcb = datasets["POS_CASH_balance"]

pcb.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10001358 entries, 0 to 10001357
Data columns (total 8 columns):
 #   Column                 Dtype
---  ------                 -----
 0   SK_ID_PREV             int64
 1   SK_ID_CURR             int64
 2   MONTHS_BALANCE         int64
 3   CNT_INSTALMENT         float64
 4   CNT_INSTALMENT_FUTURE  float64
 5   NAME_CONTRACT_STATUS   object
 6   SK_DPD                 int64
 7   SK_DPD_DEF             int64
dtypes: float64(2), int64(5), object(1)
memory usage: 610.4+ MB

pcb.head(5)

   SK_ID_PREV  SK_ID_CURR  MONTHS_BALANCE  CNT_INSTALMENT  \
0     1803195      182943             -31            48.0
1     1715348      367990             -33            36.0
2     1784872      397406             -32            12.0
3     1903291      269225             -35            48.0
4     2341044      334279             -35            36.0

   CNT_INSTALMENT_FUTURE NAME_CONTRACT_STATUS  SK_DPD  SK_DPD_DEF
```

```
0                  45.0              Active       0           0
1                  35.0              Active       0           0
2                   9.0              Active       0           0
3                  42.0              Active       0           0
4                  35.0              Active       0           0
```

```
pcb.describe()
```

```
         SK_ID_PREV    SK_ID_CURR  MONTHS_BALANCE  CNT_INSTALMENT  \
count  1.000136e+07  1.000136e+07    1.000136e+07    9.975287e+06
mean   1.903217e+06  2.784039e+05   -3.501259e+01    1.708965e+01
std    5.358465e+05  1.027637e+05    2.606657e+01    1.199506e+01
min    1.000001e+06  1.000010e+05   -9.600000e+01    1.000000e+00
25%    1.434405e+06  1.895500e+05   -5.400000e+01    1.000000e+01
50%    1.896565e+06  2.786540e+05   -2.800000e+01    1.200000e+01
75%    2.368963e+06  3.674290e+05   -1.300000e+01    2.400000e+01
max    2.843499e+06  4.562550e+05   -1.000000e+00    9.200000e+01

       CNT_INSTALMENT_FUTURE        SK_DPD    SK_DPD_DEF
count           9.975271e+06  1.000136e+07  1.000136e+07
mean            1.048384e+01  1.160693e+01  6.544684e-01
std             1.110906e+01  1.327140e+02  3.276249e+01
min             0.000000e+00  0.000000e+00  0.000000e+00
25%             3.000000e+00  0.000000e+00  0.000000e+00
50%             7.000000e+00  0.000000e+00  0.000000e+00
75%             1.400000e+01  0.000000e+00  0.000000e+00
max             8.500000e+01  4.231000e+03  3.595000e+03
```

```
app_train_pcb = pd.merge(app_train_req,
pcb[["SK_ID_CURR","SK_ID_PREV"]], how="left")
app_train_pcb.fillna(0, inplace=True)
app_train_pcb.SK_ID_PREV=app_train_pcb.SK_ID_PREV.astype(int)
```

```
display(app_train_pcb[app_train_pcb.TARGET==1].sort_values(by="SK_ID_P
REV",ascending=False).head(15))
display(app_train_pcb[app_train_pcb.TARGET==0].sort_values(by="SK_ID_P
REV",ascending=False).head(5))
```

```
         SK_ID_CURR  TARGET  SK_ID_PREV
3850632      260963       1     2843495
3850647      260963       1     2843495
3850638      260963       1     2843495
3850643      260963       1     2843495
3850644      260963       1     2843495
3850645      260963       1     2843495
3850646      260963       1     2843495
3850642      260963       1     2843495
5279022      320127       1     2843481
5279034      320127       1     2843481
5279021      320127       1     2843481
5279020      320127       1     2843481
```

```
5279014      320127       1     2843481
5279013      320127       1     2843481
5279027      320127       1     2843481

         SK_ID_CURR  TARGET  SK_ID_PREV
5135599      314148       0     2843499
5135574      314148       0     2843499
5135578      314148       0     2843499
5135591      314148       0     2843499
5135592      314148       0     2843499

pcb.SK_ID_PREV.value_counts()

1856103    96
2706683    96
1617536    96
1364606    96
1057553    96
           ..
1922777     1
2660098     1
1364218     1
1077449     1
1191779     1
Name: SK_ID_PREV, Length: 936325, dtype: int64
```

Observring Random SK_ID_PREV to gain insights

```
observation_ids = [2843495, 2843499, 2843481]
for grp, df in pcb.groupby("SK_ID_PREV"):
    if grp in observation_ids:
        display(pd.merge(df, app_train_req,
how="left").sort_values(by="MONTHS_BALANCE", ascending=False))
        observation_ids.remove(grp)
        if len(observation_ids) ==0:
            break
```

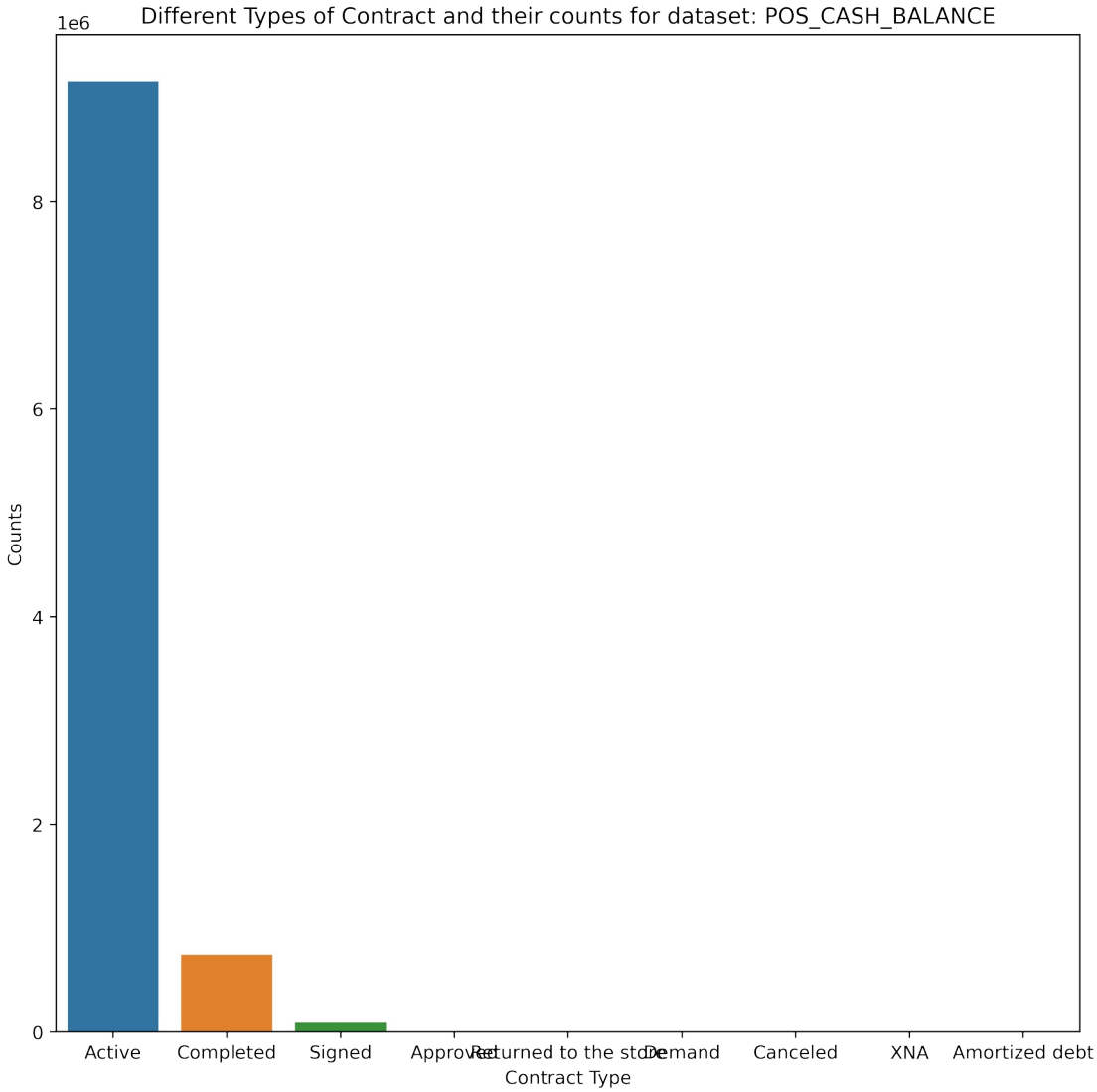
```
    SK_ID_PREV  SK_ID_CURR  MONTHS_BALANCE  CNT_INSTALMENT  \
2      2843481      320127             -78            10.0
6      2843481      320127             -79            10.0
0      2843481      320127             -80            10.0
3      2843481      320127             -81            10.0
7      2843481      320127             -82            10.0
9      2843481      320127             -83            10.0
1      2843481      320127             -84            10.0
4      2843481      320127             -85            10.0
8      2843481      320127             -86            10.0
10     2843481      320127             -87            10.0
5      2843481      320127             -88            10.0

    CNT_INSTALMENT_FUTURE NAME_CONTRACT_STATUS  SK_DPD  SK_DPD_DEF
TARGET
```

```
2                     0.0            Completed       0           0
1
6                     1.0               Active       0           0
1
0                     2.0               Active       0           0
1
3                     3.0               Active       0           0
1
7                     4.0               Active       0           0
1
9                     5.0               Active       0           0
1
1                     6.0               Active       0           0
1
4                     7.0               Active       0           0
1
8                     8.0               Active       0           0
1
10                    9.0               Active       0           0
1
5                    10.0               Active       0           0
1

   SK_ID_PREV  SK_ID_CURR  MONTHS_BALANCE  CNT_INSTALMENT  \
0     2843495      260963              -9             7.0
4     2843495      260963             -10            60.0
3     2843495      260963             -11            60.0
6     2843495      260963             -12            60.0
2     2843495      260963             -13            60.0
1     2843495      260963             -14            60.0
7     2843495      260963             -15            60.0
5     2843495      260963             -16            60.0

   CNT_INSTALMENT_FUTURE NAME_CONTRACT_STATUS  SK_DPD  SK_DPD_DEF
TARGET
0                    0.0            Completed       0           0
1
4                   54.0               Active       0           0
1
3                   55.0               Active       0           0
1
6                   56.0               Active       0           0
1
2                   57.0               Active       0           0
1
1                   58.0               Active       0           0
1
7                   59.0               Active       0           0
1
```

```
5                   60.0               Active       0           0
1

    SK_ID_PREV  SK_ID_CURR  MONTHS_BALANCE  CNT_INSTALMENT  \
8      2843499      314148             -30            10.0
5      2843499      314148             -31            10.0
1      2843499      314148             -32            60.0
2      2843499      314148             -33            60.0
7      2843499      314148             -34            60.0
0      2843499      314148             -35            60.0
10     2843499      314148             -36            60.0
6      2843499      314148             -37            60.0
9      2843499      314148             -38            60.0
4      2843499      314148             -39            60.0
3      2843499      314148             -40            60.0

    CNT_INSTALMENT_FUTURE NAME_CONTRACT_STATUS  SK_DPD  SK_DPD_DEF
TARGET
8                     0.0            Completed       0           0
0
5                     0.0               Active       0           0
0
1                    51.0               Active       0           0
0
2                    52.0               Active       0           0
0
7                    54.0               Active       0           0
0
0                    55.0               Active       0           0
0
10                   56.0               Active       0           0
0
6                    57.0               Active       0           0
0
9                    58.0               Active       0           0
0
4                    59.0               Active       0           0
0
3                    60.0               Active       0           0
0
```

```
fig, ax =  plt.subplots(1,1, figsize=(10,10), dpi=400)
sns.countplot(x='NAME_CONTRACT_STATUS', data=pcb,ax=ax)
ax.set_xlabel("Contract Type")
ax.set_ylabel("Counts")
ax.set_title("Different Types of Contract and their counts for
dataset: POS_CASH_BALANCE")
```

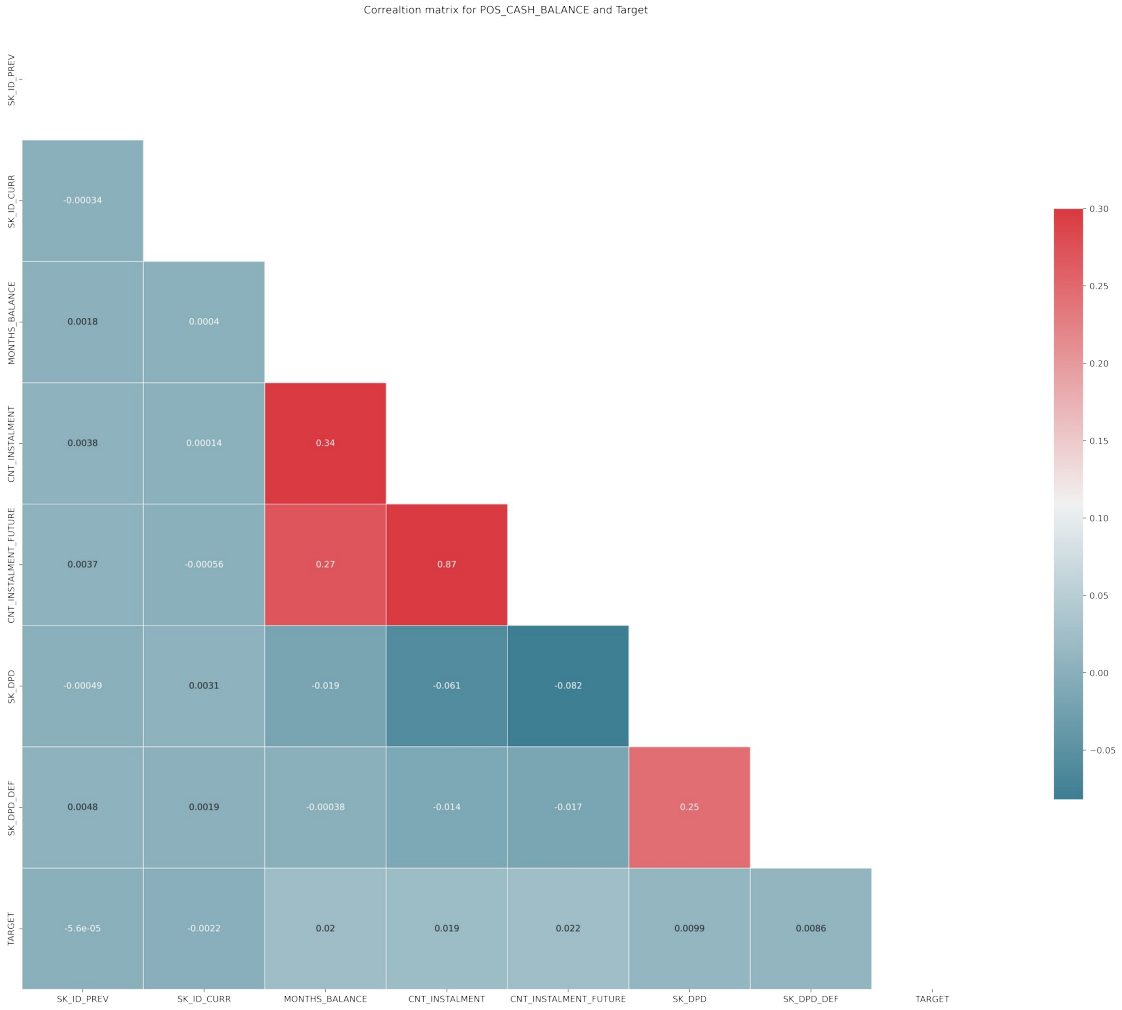
```
Text(0.5, 1.0, 'Different Types of Contract and their counts for
dataset: POS_CASH_BALANCE')
```




```
merged_df = pd.merge(pcb, app_train_req, how="left", on="SK_ID_CURR")

# set up the matplotlib figure
f, ax = plt.subplots(1,1, figsize=(25, 25), dpi=500)

# generate a mask for the lower triangle
mask = np.zeros_like(merged_df.corr(), dtype=np.bool_)
mask[np.triu_indices_from(mask)] = True

# generate a custom diverging colormap
cmap = sns.diverging_palette(220, 11, as_cmap=True)

# draw the heatmap with the mask and correct aspect ratio
sns.heatmap(merged_df.corr(), mask=mask, cmap=cmap, vmax=.3,
            square=True,
            linewidths=.5, cbar_kws={"shrink": .5}, ax=ax,
```

```
annot=True);
ax.set_title("Correaltion matrix for POS_CASH_BALANCE and Target ")
```

```
Text(0.5, 1.0, 'Correaltion matrix for POS_CASH_BALANCE and Target ')
```



Insights on aggregated data

```
agg_data =
pcb.groupby("SK_ID_PREV").agg(['mean','count','sum','min','max'])
```

```
agg_data.columns
```

```
MultiIndex([(           'SK_ID_CURR',  'mean'),
            (           'SK_ID_CURR', 'count'),
            (           'SK_ID_CURR',   'sum'),
            (           'SK_ID_CURR',   'min'),
            (           'SK_ID_CURR',   'max'),
            (       'MONTHS_BALANCE',  'mean'),
            (       'MONTHS_BALANCE', 'count'),
            (       'MONTHS_BALANCE',   'sum'),
            (       'MONTHS_BALANCE',   'min'),
```

```
            (       'MONTHS_BALANCE',   'max'),
            (       'CNT_INSTALMENT',  'mean'),
            (       'CNT_INSTALMENT', 'count'),
            (       'CNT_INSTALMENT',   'sum'),
            (       'CNT_INSTALMENT',   'min'),
            (       'CNT_INSTALMENT',   'max'),
            ('CNT_INSTALMENT_FUTURE',  'mean'),
            ('CNT_INSTALMENT_FUTURE', 'count'),
            ('CNT_INSTALMENT_FUTURE',   'sum'),
            ('CNT_INSTALMENT_FUTURE',   'min'),
            ('CNT_INSTALMENT_FUTURE',   'max'),
            (               'SK_DPD',  'mean'),
            (               'SK_DPD', 'count'),
            (               'SK_DPD',   'sum'),
            (               'SK_DPD',   'min'),
            (               'SK_DPD',   'max'),
            (          'SK_DPD_DEF',  'mean'),
            (          'SK_DPD_DEF', 'count'),
            (          'SK_DPD_DEF',   'sum'),
            (          'SK_DPD_DEF',   'min'),
            (          'SK_DPD_DEF',   'max')],
          )

print("-------------MONTHS_BALANCE--------------------")
display(agg_data.head(5)["MONTHS_BALANCE"])
print("-------------CNT_INSTALMENT--------------------")
display(agg_data.head(5)["CNT_INSTALMENT"])
print("-------------CNT_INSTALMENT_FUTURE--------------------")
display(agg_data.head(5)["CNT_INSTALMENT_FUTURE"])
print("-------------SK_DPD--------------------")
display(agg_data.head(5)["SK_DPD"])


-------------MONTHS_BALANCE--------------------

            mean  count  sum  min  max
SK_ID_PREV
1000001     -9.0      3  -27  -10   -8
1000002    -52.0      5 -260  -54  -50
1000003     -2.5      4  -10   -4   -1
1000004    -25.5      8 -204  -29  -22
1000005    -51.0     11 -561  -56  -46

-------------CNT_INSTALMENT--------------------

                mean  count    sum   min   max
SK_ID_PREV
1000001     8.666667      3   26.0   2.0  12.0
1000002     5.200000      5   26.0   4.0   6.0
1000003    12.000000      4   48.0  12.0  12.0
```

```
1000004      9.625000      8   77.0   7.0  10.0
1000005     10.000000     11  110.0  10.0  10.0

-------------CNT_INSTALMENT_FUTURE--------------------

                mean  count   sum  min   max
SK_ID_PREV
1000001      7.666667      3  23.0  0.0  12.0
1000002      2.000000      5  10.0  0.0   4.0
1000003     10.500000      4  42.0  9.0  12.0
1000004      6.125000      8  49.0  0.0  10.0
1000005      5.000000     11  55.0  0.0  10.0

-------------SK_DPD--------------------

            mean  count  sum  min  max
SK_ID_PREV
1000001      0.0      3    0    0    0
1000002      0.0      5    0    0    0
1000003      0.0      4    0    0    0
1000004      0.0      8    0    0    0
1000005      0.0     11    0    0    0
```

Dataset : installments_payment

```
ip = datasets['installments_payments']
```

```
ip.head(5)
```

```
   SK_ID_PREV  SK_ID_CURR  NUM_INSTALMENT_VERSION
NUM_INSTALMENT_NUMBER  \
0     1054186      161674                     1.0
6
1     1330831      151639                     0.0
34
2     2085231      193053                     2.0
1
3     2452527      199697                     1.0
3
4     2714724      167756                     1.0
2

   DAYS_INSTALMENT  DAYS_ENTRY_PAYMENT  AMT_INSTALMENT  AMT_PAYMENT
0          -1180.0             -1187.0        6948.360     6948.360
1          -2156.0             -2156.0        1716.525     1716.525
2            -63.0               -63.0       25425.000    25425.000
3          -2418.0             -2426.0       24350.130    24350.130
4          -1383.0             -1366.0        2165.040     2160.585
```

```
ip.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13605401 entries, 0 to 13605400
Data columns (total 8 columns):
 #   Column                  Dtype
---  ------                  -----
 0   SK_ID_PREV              int64
 1   SK_ID_CURR              int64
 2   NUM_INSTALMENT_VERSION  float64
 3   NUM_INSTALMENT_NUMBER   int64
 4   DAYS_INSTALMENT         float64
 5   DAYS_ENTRY_PAYMENT      float64
 6   AMT_INSTALMENT          float64
 7   AMT_PAYMENT             float64
dtypes: float64(5), int64(3)
memory usage: 830.4 MB
```

```
ip.describe()
```

```
         SK_ID_PREV    SK_ID_CURR  NUM_INSTALMENT_VERSION  \
count  1.360540e+07  1.360540e+07            1.360540e+07
mean   1.903365e+06  2.784449e+05            8.566373e-01
std    5.362029e+05  1.027183e+05            1.035216e+00
min    1.000001e+06  1.000010e+05            0.000000e+00
25%    1.434191e+06  1.896390e+05            0.000000e+00
50%    1.896520e+06  2.786850e+05            1.000000e+00
75%    2.369094e+06  3.675300e+05            1.000000e+00
max    2.843499e+06  4.562550e+05            1.780000e+02

       NUM_INSTALMENT_NUMBER  DAYS_INSTALMENT  DAYS_ENTRY_PAYMENT  \
count           1.360540e+07     1.360540e+07        1.360250e+07
mean            1.887090e+01    -1.042270e+03       -1.051114e+03
std             2.666407e+01     8.009463e+02        8.005859e+02
min             1.000000e+00    -2.922000e+03       -4.921000e+03
25%             4.000000e+00    -1.654000e+03       -1.662000e+03
50%             8.000000e+00    -8.180000e+02       -8.270000e+02
75%             1.900000e+01    -3.610000e+02       -3.700000e+02
max             2.770000e+02    -1.000000e+00       -1.000000e+00

       AMT_INSTALMENT   AMT_PAYMENT
count    1.360540e+07  1.360250e+07
mean     1.705091e+04  1.723822e+04
std      5.057025e+04  5.473578e+04
min      0.000000e+00  0.000000e+00
25%      4.226085e+03  3.398265e+03
50%      8.884080e+03  8.125515e+03
75%      1.671021e+04  1.610842e+04
max      3.771488e+06  3.771488e+06
```

```
ip.isna().sum()
```

```
SK_ID_PREV                   0
SK_ID_CURR                   0
NUM_INSTALMENT_VERSION       0
NUM_INSTALMENT_NUMBER        0
DAYS_INSTALMENT              0
DAYS_ENTRY_PAYMENT        2905
AMT_INSTALMENT               0
AMT_PAYMENT               2905
dtype: int64
```

```
app_train = datasets['application_train']
app_train_req = app_train[["SK_ID_CURR", "TARGET"]]


app_train_ip = pd.merge(app_train_req,
ip[["SK_ID_CURR","SK_ID_PREV"]], how="left")
app_train_ip.fillna(0, inplace=True)
app_train_ip.SK_ID_PREV=app_train_ip.SK_ID_PREV.astype(int)


display(app_train_ip[app_train_ip.TARGET==1].sort_values(by="SK_ID_PRE
V",ascending=False).head(5))
display(app_train_ip[app_train_ip.TARGET==0].sort_values(by="SK_ID_PRE
V",ascending=False).head(5))
```

```
         SK_ID_CURR  TARGET  SK_ID_PREV
5219784      260963       1     2843495
5219782      260963       1     2843495
5219789      260963       1     2843495
5219788      260963       1     2843495
5219779      260963       1     2843495

         SK_ID_CURR  TARGET  SK_ID_PREV
6957373      314148       0     2843499
6957375      314148       0     2843499
6957389      314148       0     2843499
6957388      314148       0     2843499
6957382      314148       0     2843499
```

Observring Random SK_ID_PREV to gain insights

```
observation_ids = [2843499, 2843495, 2843461]
for grp, df in ip.groupby("SK_ID_PREV"):
    if grp in observation_ids:
        display(pd.merge(df, app_train_req,
how="left").sort_values(by="NUM_INSTALMENT_NUMBER"))
        observation_ids.remove(grp)
        if len(observation_ids) ==0:
            break
```

```
    SK_ID_PREV  SK_ID_CURR  NUM_INSTALMENT_VERSION
NUM_INSTALMENT_NUMBER  \
```

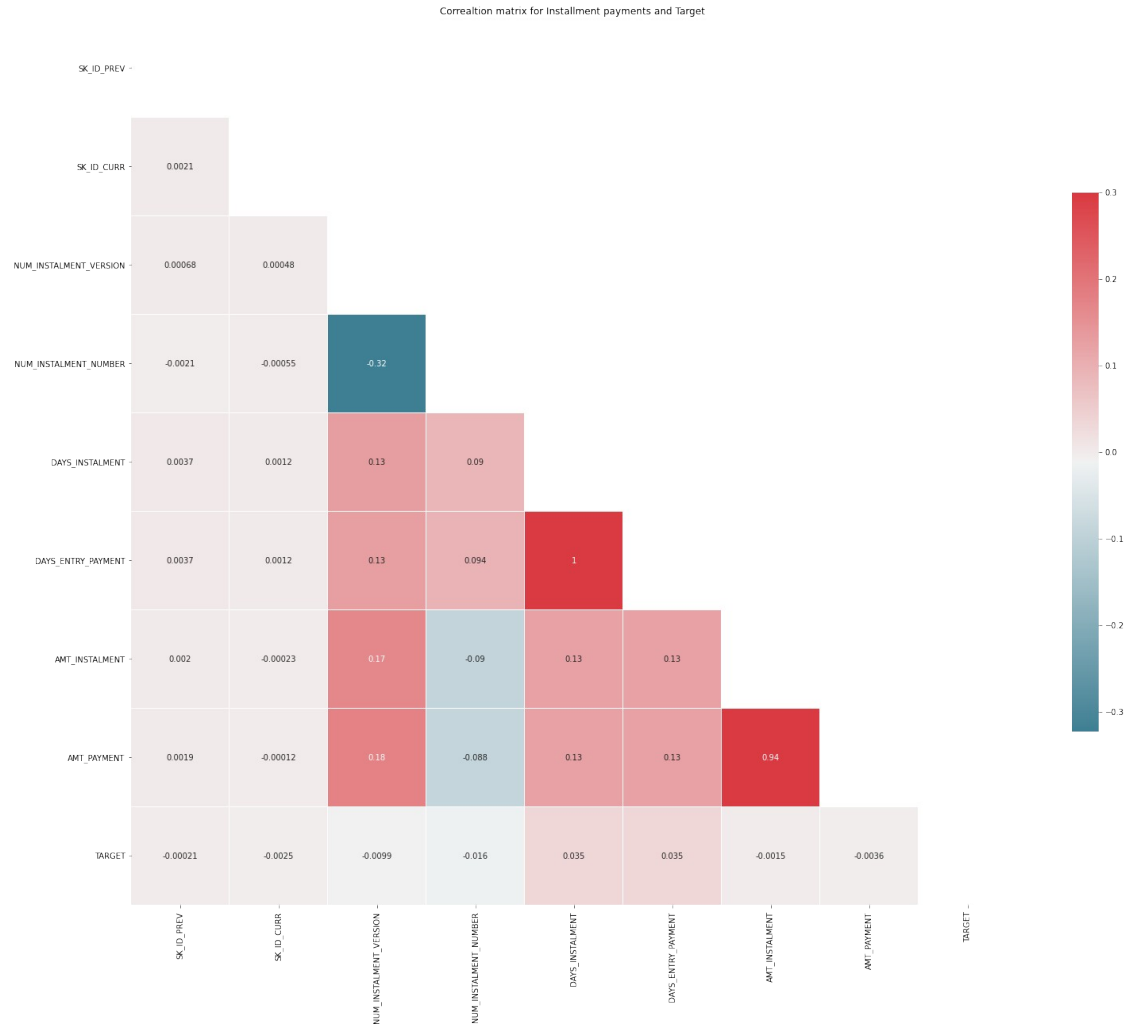
```
30     2843461      210644                     0.0
1
32     2843461      210644                     0.0
2
56     2843461      210644                     0.0
3
1      2843461      210644                     0.0
4
24     2843461      210644                     0.0
5
..         ...         ...                     ...
...
21     2843461      210644                     0.0
95
70     2843461      210644                     0.0
96
27     2843461      210644                     0.0
97
64     2843461      210644                     0.0
98
26     2843461      210644                     0.0
99

    DAYS_INSTALMENT  DAYS_ENTRY_PAYMENT  AMT_INSTALMENT  AMT_PAYMENT
TARGET
30          -2187.0             -2216.0          4500.0       4500.0
1
32          -2200.0             -2200.0          9000.0       9000.0
1
56          -2156.0             -2155.0          4500.0       4500.0
1
1           -2126.0             -2155.0          4500.0       4500.0
1
24          -2155.0             -2155.0          4500.0       4500.0
1
..              ...                 ...             ...          ...
...
21            -88.0               -93.0          2250.0       2250.0
1
70            -93.0               -93.0           450.0        450.0
1
27            -57.0               -63.0          2250.0       2250.0
1
64            -26.0               -39.0          2250.0       2250.0
1
26            -39.0               -39.0           450.0        450.0
1

[99 rows x 9 columns]
```

```
   SK_ID_PREV  SK_ID_CURR  NUM_INSTALMENT_VERSION
NUM_INSTALMENT_NUMBER  \
0     2843495      260963                     1.0
1
3     2843495      260963                     1.0
2
4     2843495      260963                     1.0
3
2     2843495      260963                     1.0
4
5     2843495      260963                     1.0
5
1     2843495      260963                     1.0
6
6     2843495      260963                     2.0
7

   DAYS_INSTALMENT  DAYS_ENTRY_PAYMENT  AMT_INSTALMENT  AMT_PAYMENT
TARGET
0            -439.0              -453.0       23556.195    23556.195
1
3            -409.0              -412.0       23556.195    23556.195
1
4            -379.0              -384.0       23556.195    23556.195
1
2            -349.0              -349.0       23556.195    23556.195
1
5            -319.0              -320.0       23556.195    23556.195
1
1            -289.0              -288.0       23556.195    23556.195
1
6            -259.0              -264.0      656193.015   656193.015
1

   SK_ID_PREV  SK_ID_CURR  NUM_INSTALMENT_VERSION
NUM_INSTALMENT_NUMBER  \
4     2843499      314148                     1.0
1
2     2843499      314148                     1.0
2
9     2843499      314148                     1.0
3
3     2843499      314148                     1.0
4
7     2843499      314148                     1.0
5
8     2843499      314148                     1.0
6
0     2843499      314148                     1.0
7
```

```
5     2843499      314148                    1.0
8
6     2843499      314148                    1.0
9
1     2843499      314148                    2.0
10

   DAYS_INSTALMENT  DAYS_ENTRY_PAYMENT  AMT_INSTALMENT  AMT_PAYMENT
TARGET
4          -1203.0             -1220.0        16074.00     16074.00
0
2          -1173.0             -1201.0        16074.00     16074.00
0
9          -1143.0             -1172.0        16074.00     16074.00
0
3          -1113.0             -1138.0        16074.00     16074.00
0
7          -1083.0             -1109.0        16074.00     16074.00
0
8          -1053.0             -1074.0        16074.00     16074.00
0
0          -1023.0             -1047.0        16074.00     16074.00
0
5           -993.0             -1018.0        16074.00     16074.00
0
6           -963.0              -980.0        16074.00     16074.00
0
1           -933.0              -952.0       433416.24    433416.24
0
```

```
merged_df = pd.merge(ip, app_train_req, how="left", on="SK_ID_CURR")

# set up the matplotlib figure
f, ax = plt.subplots(1,1, figsize=(25, 25))

# generate a mask for the lower triangle
mask = np.zeros_like(merged_df.corr(), dtype=np.bool_)
mask[np.triu_indices_from(mask)] = True

# generate a custom diverging colormap
cmap = sns.diverging_palette(220, 11, as_cmap=True)

# draw the heatmap with the mask and correct aspect ratio
sns.heatmap(merged_df.corr(), mask=mask, cmap=cmap, vmax=.3,
            square=True,
            linewidths=.5, cbar_kws={"shrink": .5}, ax=ax,
annot=True);
ax.set_title("Correaltion matrix for Installment payments and Target
")
```

```
Text(0.5, 1.0, 'Correaltion matrix for Installment payments and Target
')
```




```
ip.NUM_INSTALMENT_VERSION.unique()
```

```
array([  1.,   0.,   2.,   4.,   3.,   5.,   7.,   8.,   6.,  13.,
9.,
       21.,  22.,  12.,  17.,  18.,  11.,  14.,  34.,  33.,  19.,
16.,
       15.,  10.,  26.,  27.,  20.,  25.,  23.,  24.,  31.,  32.,
28.,
       35.,  29.,  30.,  43.,  39.,  40.,  36.,  41.,  42.,  37.,
38.,
       68.,  44.,  45.,  46., 178.,  52.,  51.,  53.,  54.,  49.,
50.,
       58.,  57.,  55.,  56.,  48.,  47.,  72.,  59.,  73.,  61.])
```

Aggreating data.

```
agg_data =
ip.groupby("SK_ID_PREV").agg(['mean','count','sum','min','max'])
```

```
agg_data.columns
```

```
MultiIndex([(            'SK_ID_CURR',  'mean'),
            (            'SK_ID_CURR', 'count'),
            (            'SK_ID_CURR',   'sum'),
            (            'SK_ID_CURR',   'min'),
            (            'SK_ID_CURR',   'max'),
            ('NUM_INSTALMENT_VERSION',  'mean'),
            ('NUM_INSTALMENT_VERSION', 'count'),
            ('NUM_INSTALMENT_VERSION',   'sum'),
            ('NUM_INSTALMENT_VERSION',   'min'),
            ('NUM_INSTALMENT_VERSION',   'max'),
            ( 'NUM_INSTALMENT_NUMBER',  'mean'),
            ( 'NUM_INSTALMENT_NUMBER', 'count'),
            ( 'NUM_INSTALMENT_NUMBER',   'sum'),
            ( 'NUM_INSTALMENT_NUMBER',   'min'),
            ( 'NUM_INSTALMENT_NUMBER',   'max'),
            (       'DAYS_INSTALMENT',  'mean'),
            (       'DAYS_INSTALMENT', 'count'),
            (       'DAYS_INSTALMENT',   'sum'),
            (       'DAYS_INSTALMENT',   'min'),
            (       'DAYS_INSTALMENT',   'max'),
            (    'DAYS_ENTRY_PAYMENT',  'mean'),
            (    'DAYS_ENTRY_PAYMENT', 'count'),
            (    'DAYS_ENTRY_PAYMENT',   'sum'),
            (    'DAYS_ENTRY_PAYMENT',   'min'),
            (    'DAYS_ENTRY_PAYMENT',   'max'),
            (        'AMT_INSTALMENT',  'mean'),
            (        'AMT_INSTALMENT', 'count'),
            (        'AMT_INSTALMENT',   'sum'),
            (        'AMT_INSTALMENT',   'min'),
            (        'AMT_INSTALMENT',   'max'),
            (           'AMT_PAYMENT',  'mean'),
            (           'AMT_PAYMENT', 'count'),
            (           'AMT_PAYMENT',   'sum'),
            (           'AMT_PAYMENT',   'min'),
            (           'AMT_PAYMENT',   'max')],
           )
```

```
agg_data.head(5)["AMT_INSTALMENT"]
```

SK_ID_PREV	mean	count	sum	min	max
1000001	34221.712500	2	68443.425	6404.31	62039.115
1000002	9308.891250	4	37235.565	6264.00	18443.565
1000003	4951.350000	3	14854.050	4951.35	4951.350
1000004	4789.022143	7	33523.155	3391.11	13176.495
1000005	14703.210000	11	161735.310	14599.26	14713.605

```
agg_data.head(5)["AMT_PAYMENT"]
```

```
                    mean  count         sum      min        max
SK_ID_PREV
1000001     34221.712500      2   68443.425  6404.31  62039.115
1000002      9308.891250      4   37235.565  6264.00  18443.565
1000003      4951.350000      3   14854.050  4951.35   4951.350
1000004      4789.022143      7   33523.155  3391.11  13176.495
1000005     13365.609545     11  147021.705     2.79  14713.605
```

- [ ] Completed???

Dataset questions

Unique record for each SK_ID_CURR

```
datasets.keys()
```

```
dict_keys(['application_train', 'application_test', 'bureau',
'bureau_balance', 'credit_card_balance', 'installments_payments',
'previous_application', 'POS_CASH_balance'])
```

```
len(datasets["application_train"]["SK_ID_CURR"].unique()) ==
datasets["application_train"].shape[0]
```

```
True
```

```
np.intersect1d(datasets["application_train"]["SK_ID_CURR"],
datasets["application_test"]["SK_ID_CURR"])
```

```
array([], dtype=int64)
```

```
datasets["application_test"].shape
```

```
(48744, 121)
```

```
datasets["application_train"].shape
```

```
(160892, 122)
```

Previous applications for the submission file

The persons in the kaggle submission file have had previous applications in the `previous_application.csv`. 47,800 out 48,744 people have had previous appications.

```
appsDF = datasets["previous_application"]
```

```
appsDF.shape
```

```
(1670214, 37)
```

```
len(np.intersect1d(datasets["previous_application"]["SK_ID_CURR"],
datasets["application_test"]["SK_ID_CURR"]))
```

```
47800
```

```
print(f"There are  {appsDF.shape[0]:,} previous applications")
```

```
There are  1,670,214 previous applications
```

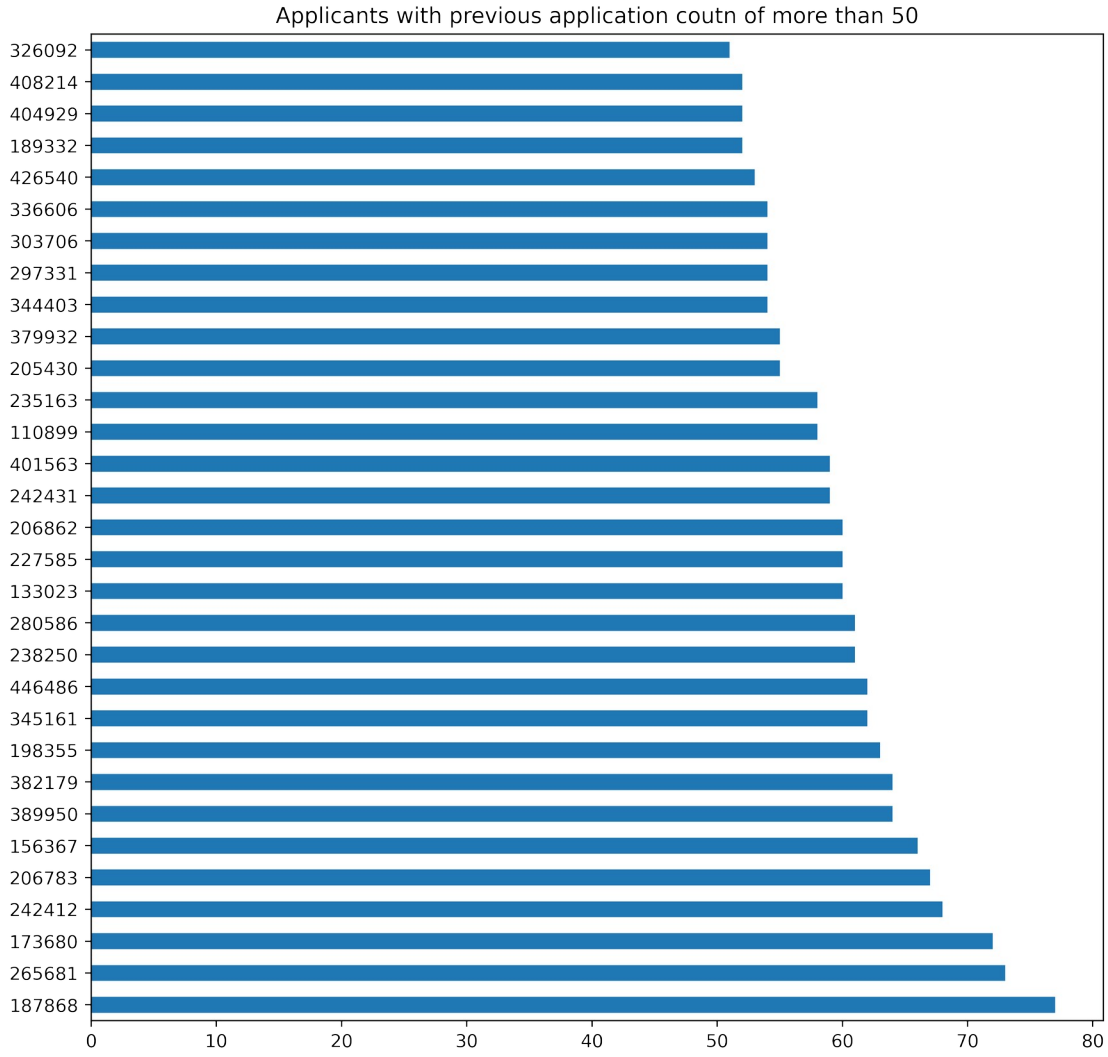
```
# How many entries are there for each month?
prevAppCounts = appsDF['SK_ID_CURR'].value_counts(dropna=False)
```

```
len(prevAppCounts[prevAppCounts >40])  #more that 40 previous
applications
```

```
101
```

```
fig, ax = plt.subplots(1,1, figsize=(10,10), dpi=400)
prevAppCounts[prevAppCounts >50].plot(kind='barh', ax=ax)
ax.set_title("Applicants with previous application coutn of more than
50")
```

```
Text(0.5, 1.0, 'Applicants with previous application coutn of more
than 50')
```



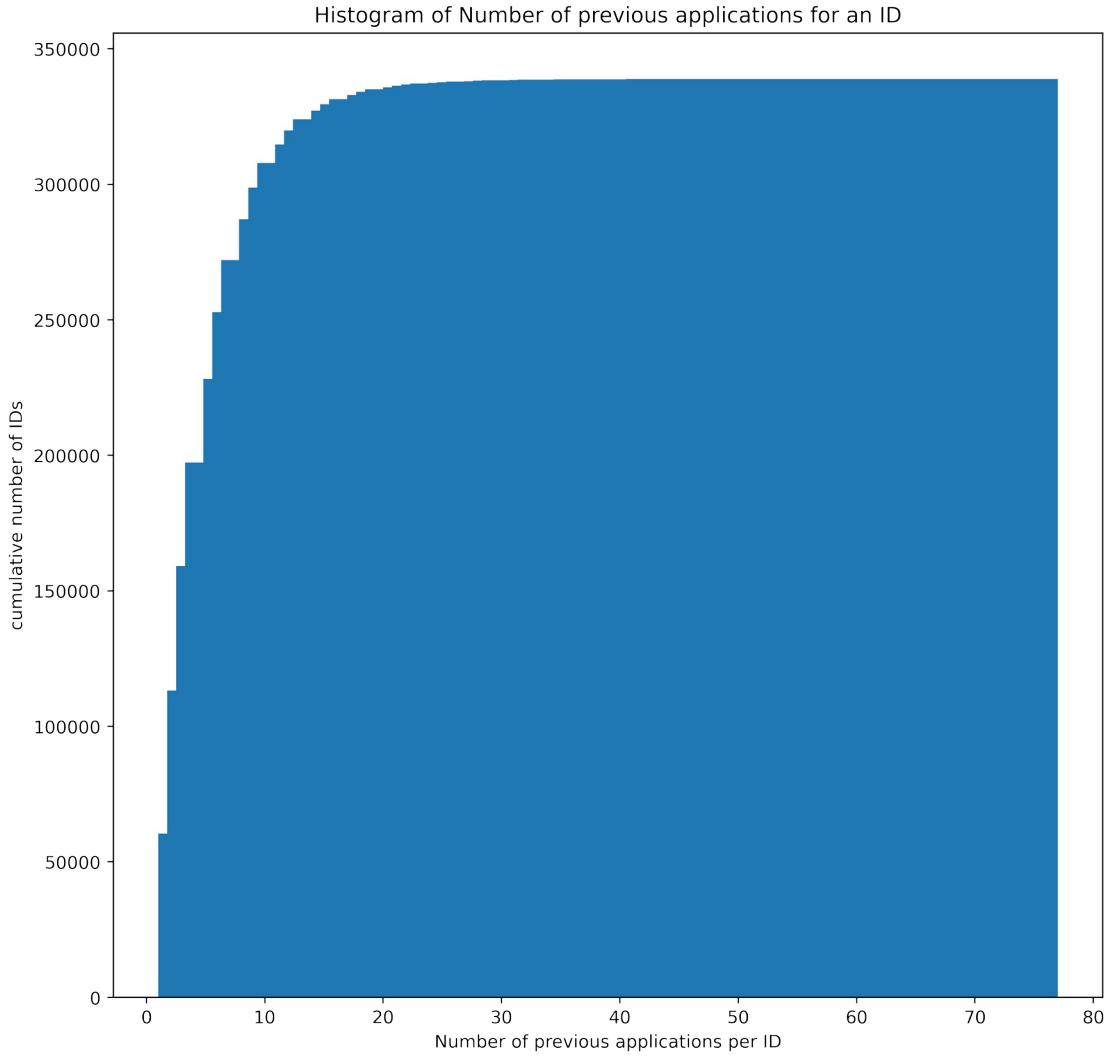
Histogram of Number of previous applications for an ID

```
sum(appsDF['SK_ID_CURR'].value_counts()==1)
```

```
60458
```

```
fig, ax = plt.subplots(1,1, figsize=(10,10), dpi=400)
plt.hist(appsDF['SK_ID_CURR'].value_counts(), cumulative =True, bins =
100, histtype="barstacked")
ax.set_ylabel('cumulative number of IDs')
ax.set_xlabel('Number of previous applications per ID')
ax.set_title('Histogram of Number of previous applications for an ID')
```

```
Text(0.5, 1.0, 'Histogram of Number of previous applications for an
ID')
```



Can we differentiate applications by low, medium and high previous apps?

* Low = <5 claims (22%)
* Medium = 10 to 39 claims (58%)
* High = 40 or more claims (20%)

```
apps_all = appsDF['SK_ID_CURR'].nunique()
apps_5plus = appsDF['SK_ID_CURR'].value_counts()<=5
apps_40plus = appsDF['SK_ID_CURR'].value_counts()>=40
print('Percentage with 5 or less previous apps:',
np.round(100.*(sum(apps_5plus)/apps_all),5))
print('Percentage with 40 or more previous apps:',
np.round(100.*(sum(apps_40plus)/apps_all),5))
```

```
Percentage with 5 or less previous apps: 67.34581
Percentage with 40 or more previous apps: 0.03453
```

Design Decisions | Sample Examples | Sample Feature Engineering

Joining secondary tables with the primary table

In the case of the HCDR competition (and many other machine learning problems that involve multiple tables in 3NF or not) we need to join these datasets (denormalize) when using a machine learning pipeline. Joining the secondary tables with the primary table will lead to lots of new features about each loan application; these features will tend to be aggregate type features or meta data about the loan or its application. How can we do this when using Machine Learning Pipelines?

Joining `previous_application` with `application_x`

We refer to the `application_train` data (and also `application_test` data also) as the **primary table** and the other files as the **secondary tables** (e.g., `previous_application` dataset). All tables can be joined using the primary key `SK_ID_PREV`.

Let's assume we wish to generate a feature based on previous application attempts. In this case, possible features here could be:

- A simple feature could be the number of previous applications.
- Other summary features of original features such as `AMT_APPLICATION`, `AMT_CREDIT` could be based on average, min, max, median, etc.

To build such features, we need to join the `application_train` data (and also `application_test` data also) with the 'previous_application' dataset (and the other available datasets).

When joining this data in the context of pipelines, different strategies come to mind with various tradeoffs:

1. Preprocess each of the non-application data sets, thereby generating many new (derived) features, and then joining (aka merge) the results with the `application_train` data (the labeled dataset) and with the `application_test` data (the unlabeled submission dataset) prior to processing the data (in a train, valid, test partition) via your machine learning pipeline. [This approach is recommended for this HCDR competition. WHY?]
- Do the joins as part of the transformation steps. [Not recommended here. WHY?]. How can this be done? Will it work?
    - This would be necessary if we had dataset wide features such as IDF (inverse document frequency) which depend on the entire subset of data as opposed to a single loan application (e.g., a feature about the relative amount applied for such as the percentile of the loan amount being applied for).

I want you to think about this section and build on this.

Roadmap for secondary table processing

1. Transform all the secondary tables to features that can be joined into the main table the application table (labeled and unlabeled)
   - 'bureau', 'bureau_balance', 'credit_card_balance', 'installments_payments',
   - 'previous_application', 'POS_CASH_balance'

- Merge the transformed secondary tables with the primary tables (i.e., the `application_train` data (the labeled dataset) and with the `application_test` data (the unlabeled submission dataset)), thereby leading to X_train, y_train, X_valid, etc.
- Proceed with the learning pipeline using X_train, y_train, X_valid, etc.
- Generate a submission file using the learnt model

Pandas dataframe aggegration detour

Aggregate using one or more operations over the specified axis.

For more details see agg

```
DataFrame.agg(func, axis=0, *args, **kwargs**)
```

Aggregate using one or more operations over the specified axis.

```
df = pd.DataFrame([[1, 2, 3],
                   [4, 5, 6],
                  [7, 8, 9],
                  [np.nan, np.nan, np.nan]],
                  columns=['A', 'B', 'C'])

df
```

```
     A    B    C
0  1.0  2.0  3.0
1  4.0  5.0  6.0
2  7.0  8.0  9.0
3  NaN  NaN  NaN
```

```
df.agg({'A' : ['sum', 'min'], 'B' : ['min', 'max']})
#        A    B
#max   NaN  8.0
#min   1.0  2.0
#sum  12.0  NaN
```

```
        A    B
sum  12.0  NaN
min   1.0  2.0
max   NaN  8.0
```

```
df = pd.DataFrame({'A': [1, 1, 2, 2],
                   'B': [1, 2, 3, 4],
```

```
                    'C': np.random.randn(4)})
df
```

```
   A  B         C
0  1  1  1.659589
1  1  2  0.377106
2  2  3 -1.816287
3  2  4 -0.390450
```

```
df.groupby('A').agg({'B': ['min', 'max'], 'C': 'sum'})
#    B             C
#  min max       sum
#A
#1   1   2  0.590716
#2   3   4  0.704907
```

```
    B             C
  min max       sum
A
1   1   2  2.036695
2   3   4 -2.206736
```

```
funcs = ["a","b","c"]
{f:f"{f}_max" for f in funcs}
```

```
{'a': 'a_max', 'b': 'b_max', 'c': 'c_max'}
```

Multiple condition expressions in Pandas

So far, both our boolean selections have involved a single condition. You can, of course, have as many conditions as you would like. To do so, you will need to combine your boolean expressions using the three logical operators and, or and not.

Use &, | , ~ Although Python uses the syntax and, or, and not, these will not work when testing multiple conditions with pandas. The details of why are explained here.

You must use the following operators with pandas:

- & for and
- | for or
- ~ for not

```
appsDF[0:50][(appsDF["SK_ID_CURR"]==175704)]
```

```
   SK_ID_PREV  SK_ID_CURR NAME_CONTRACT_TYPE  AMT_ANNUITY
AMT_APPLICATION  \
6     2315218      175704         Cash loans          NaN
0.0

   AMT_CREDIT  AMT_DOWN_PAYMENT  AMT_GOODS_PRICE
WEEKDAY_APPR_PROCESS_START  \
6         0.0               NaN              NaN
```

```
TUESDAY

   HOUR_APPR_PROCESS_START  ... NAME_SELLER_INDUSTRY  CNT_PAYMENT  \
6                       11  ...                 XNA          NaN

   NAME_YIELD_GROUP  PRODUCT_COMBINATION  DAYS_FIRST_DRAWING
DAYS_FIRST_DUE  \
6              XNA                 Cash                 NaN
NaN

  DAYS_LAST_DUE_1ST_VERSION  DAYS_LAST_DUE DAYS_TERMINATION  \
6                       NaN            NaN              NaN

  NFLAG_INSURED_ON_APPROVAL
6                       NaN

[1 rows x 37 columns]
```

```
appsDF[0:50][(appsDF["SK_ID_CURR"]==175704) &
~(appsDF["AMT_CREDIT"]==1.0)]
```

```
   SK_ID_PREV  SK_ID_CURR NAME_CONTRACT_TYPE  AMT_ANNUITY
AMT_APPLICATION  \
6     2315218      175704         Cash loans          NaN
0.0

   AMT_CREDIT  AMT_DOWN_PAYMENT  AMT_GOODS_PRICE
WEEKDAY_APPR_PROCESS_START  \
6         0.0               NaN              NaN
TUESDAY

   HOUR_APPR_PROCESS_START  ... NAME_SELLER_INDUSTRY  CNT_PAYMENT  \
6                       11  ...                 XNA          NaN

   NAME_YIELD_GROUP  PRODUCT_COMBINATION  DAYS_FIRST_DRAWING
DAYS_FIRST_DUE  \
6              XNA                 Cash                 NaN
NaN

  DAYS_LAST_DUE_1ST_VERSION  DAYS_LAST_DUE DAYS_TERMINATION  \
6                       NaN            NaN              NaN

  NFLAG_INSURED_ON_APPROVAL
6                       NaN

[1 rows x 37 columns]
```

Sample Feature Engineering | previous_applications

Missing value Analayis

```
appsDF.isna().sum()
```

```
SK_ID_PREV                        0
SK_ID_CURR                        0
NAME_CONTRACT_TYPE                0
AMT_ANNUITY                  372235
AMT_APPLICATION                   0
AMT_CREDIT                        1
AMT_DOWN_PAYMENT             895844
AMT_GOODS_PRICE              385515
WEEKDAY_APPR_PROCESS_START        0
HOUR_APPR_PROCESS_START           0
FLAG_LAST_APPL_PER_CONTRACT       0
NFLAG_LAST_APPL_IN_DAY            0
RATE_DOWN_PAYMENT            895844
RATE_INTEREST_PRIMARY       1664263
RATE_INTEREST_PRIVILEGED    1664263
NAME_CASH_LOAN_PURPOSE            0
NAME_CONTRACT_STATUS              0
DAYS_DECISION                     0
NAME_PAYMENT_TYPE                 0
CODE_REJECT_REASON                0
NAME_TYPE_SUITE              820405
NAME_CLIENT_TYPE                  0
NAME_GOODS_CATEGORY               0
NAME_PORTFOLIO                    0
NAME_PRODUCT_TYPE                 0
CHANNEL_TYPE                      0
SELLERPLACE_AREA                  0
NAME_SELLER_INDUSTRY              0
CNT_PAYMENT                  372230
NAME_YIELD_GROUP                  0
PRODUCT_COMBINATION             346
DAYS_FIRST_DRAWING           673065
DAYS_FIRST_DUE               673065
DAYS_LAST_DUE_1ST_VERSION    673065
DAYS_LAST_DUE                673065
DAYS_TERMINATION             673065
NFLAG_INSURED_ON_APPROVAL    673065
dtype: int64
```

```
appsDF.columns
```

```
Index(['SK_ID_PREV', 'SK_ID_CURR', 'NAME_CONTRACT_TYPE',
'AMT_ANNUITY',
       'AMT_APPLICATION', 'AMT_CREDIT', 'AMT_DOWN_PAYMENT',
'AMT_GOODS_PRICE',
       'WEEKDAY_APPR_PROCESS_START', 'HOUR_APPR_PROCESS_START',
```

```
       'FLAG_LAST_APPL_PER_CONTRACT', 'NFLAG_LAST_APPL_IN_DAY',
       'RATE_DOWN_PAYMENT', 'RATE_INTEREST_PRIMARY',
       'RATE_INTEREST_PRIVILEGED', 'NAME_CASH_LOAN_PURPOSE',
       'NAME_CONTRACT_STATUS', 'DAYS_DECISION', 'NAME_PAYMENT_TYPE',
       'CODE_REJECT_REASON', 'NAME_TYPE_SUITE', 'NAME_CLIENT_TYPE',
       'NAME_GOODS_CATEGORY', 'NAME_PORTFOLIO', 'NAME_PRODUCT_TYPE',
       'CHANNEL_TYPE', 'SELLERPLACE_AREA', 'NAME_SELLER_INDUSTRY',
       'CNT_PAYMENT', 'NAME_YIELD_GROUP', 'PRODUCT_COMBINATION',
       'DAYS_FIRST_DRAWING', 'DAYS_FIRST_DUE',
'DAYS_LAST_DUE_1ST_VERSION',
       'DAYS_LAST_DUE', 'DAYS_TERMINATION',
'NFLAG_INSURED_ON_APPROVAL'],
      dtype='object')
```

Sample Feature Enginnering | `Previous_application` analysis

appsDF

```
         SK_ID_PREV  SK_ID_CURR NAME_CONTRACT_TYPE  AMT_ANNUITY  \
0           2030495      271877     Consumer loans     1730.430
1           2802425      108129         Cash loans    25188.615
2           2523466      122040         Cash loans    15060.735
3           2819243      176158         Cash loans    47041.335
4           1784265      202054         Cash loans    31924.395
...             ...         ...                ...          ...
1670209     2300464      352015     Consumer loans    14704.290
1670210     2357031      334635     Consumer loans     6622.020
1670211     2659632      249544     Consumer loans    11520.855
1670212     2785582      400317         Cash loans    18821.520
1670213     2418762      261212         Cash loans    16431.300

         AMT_APPLICATION  AMT_CREDIT  AMT_DOWN_PAYMENT
AMT_GOODS_PRICE  \
0                17145.0     17145.0               0.0
17145.0
1               607500.0    679671.0               NaN
607500.0
2               112500.0    136444.5               NaN
112500.0
3               450000.0    470790.0               NaN
450000.0
4               337500.0    404055.0               NaN
337500.0
...                  ...         ...               ...              ..
.
1670209         267295.5    311400.0               0.0
267295.5
1670210          87750.0     64291.5           29250.0
87750.0
1670211         105237.0    102523.5           10525.5
105237.0
```

```
1670212         180000.0    191880.0               NaN
180000.0
1670213         360000.0    360000.0               NaN
360000.0

        WEEKDAY_APPR_PROCESS_START  HOUR_APPR_PROCESS_START  ...  \
0                         SATURDAY                       15  ...
1                         THURSDAY                       11  ...
2                          TUESDAY                       11  ...
3                           MONDAY                        7  ...
4                         THURSDAY                        9  ...
...                            ...                      ...  ...
1670209                 WEDNESDAY                       12  ...
1670210                   TUESDAY                       15  ...
1670211                    MONDAY                       12  ...
1670212                 WEDNESDAY                        9  ...
1670213                    SUNDAY                       10  ...

         NAME_SELLER_INDUSTRY  CNT_PAYMENT  NAME_YIELD_GROUP  \
0                Connectivity         12.0            middle
1                         XNA         36.0       low_action
2                         XNA         12.0              high
3                         XNA         12.0            middle
4                         XNA         24.0              high
...                       ...          ...               ...
1670209             Furniture         30.0       low_normal
1670210             Furniture         12.0            middle
1670211  Consumer electronics         10.0       low_normal
1670212                   XNA         12.0       low_normal
1670213                   XNA         48.0            middle

                 PRODUCT_COMBINATION  DAYS_FIRST_DRAWING
DAYS_FIRST_DUE  \
0          POS mobile with interest           365243.0          -
42.0
1                  Cash X-Sell: low           365243.0         -
134.0
2                Cash X-Sell: high            365243.0         -
271.0
3               Cash X-Sell: middle           365243.0         -
482.0
4                Cash Street: high                 NaN
NaN
...                             ...                ...           ..
.
1670209   POS industry with interest           365243.0         -
508.0
1670210   POS industry with interest           365243.0        -
1604.0
1670211  POS household with interest           365243.0        -
```

```
1457.0
1670212             Cash X-Sell: low            365243.0        -
1155.0
1670213          Cash X-Sell: middle            365243.0        -
1163.0

        DAYS_LAST_DUE_1ST_VERSION  DAYS_LAST_DUE DAYS_TERMINATION  \
0                           300.0          -42.0            -37.0
1                           916.0       365243.0         365243.0
2                            59.0       365243.0         365243.0
3                          -152.0         -182.0           -177.0
4                             NaN            NaN              NaN
...                           ...            ...              ...
1670209                     362.0         -358.0           -351.0
1670210                   -1274.0        -1304.0          -1297.0
1670211                   -1187.0        -1187.0          -1181.0
1670212                    -825.0         -825.0           -817.0
1670213                     247.0         -443.0           -423.0

        NFLAG_INSURED_ON_APPROVAL
0                             0.0
1                             1.0
2                             1.0
3                             1.0
4                             NaN
...                           ...
1670209                       0.0
1670210                       0.0
1670211                       0.0
1670212                       1.0
1670213                       0.0

[1670214 rows x 37 columns]
```

```
features = ['AMT_ANNUITY', 'AMT_APPLICATION']
agg_op_features = {}
for f in features: #build agg dictionary
    agg_op_features[f]=[]
    agg_op_features[f].extend((f"{f}_{func}",func) for func in ["min",
"max", "mean"])
print(f"{appsDF[features].describe()}")
print("\n\n\n Required Features...")
print(agg_op_features)
result = appsDF.groupby(["SK_ID_CURR"]).agg(agg_op_features)
result.columns = result.columns.droplevel() #drop 1 of the header row
but keep the feature name header row
result = result.reset_index(level=["SK_ID_CURR"])
result['range_AMT_APPLICATION'] = result['AMT_APPLICATION_max'] -
result['AMT_APPLICATION_min']
```

```
print(f"---------------------\n\n\n result.shape: {result.shape}")
display(result.head(10))
```

	AMT_ANNUITY	AMT_APPLICATION
count	1.297979e+06	1.670214e+06
mean	1.595512e+04	1.752339e+05
std	1.478214e+04	2.927798e+05
min	0.000000e+00	0.000000e+00
25%	6.321780e+03	1.872000e+04
50%	1.125000e+04	7.104600e+04
75%	2.065842e+04	1.803600e+05
max	4.180581e+05	6.905160e+06

```
 Required Features...
{'AMT_ANNUITY': [('AMT_ANNUITY_min', 'min'), ('AMT_ANNUITY_max',
'max'), ('AMT_ANNUITY_mean', 'mean')], 'AMT_APPLICATION':
[('AMT_APPLICATION_min', 'min'), ('AMT_APPLICATION_max', 'max'),
('AMT_APPLICATION_mean', 'mean')]}
---------------------


 result.shape: (338857, 8)
```

	SK_ID_CURR	AMT_ANNUITY_min	AMT_ANNUITY_max	AMT_ANNUITY_mean
0	100001	3951.000	3951.000	3951.000000
1	100002	9251.775	9251.775	9251.775000
2	100003	6737.310	98356.995	56553.990000
3	100004	5357.250	5357.250	5357.250000
4	100005	4813.200	4813.200	4813.200000
5	100006	2482.920	39954.510	23651.175000
6	100007	1834.290	22678.785	12278.805000
7	100008	8019.090	25309.575	15839.696250
8	100009	7435.845	17341.605	10051.412143
9	100010	27463.410	27463.410	27463.410000

	AMT_APPLICATION_min	AMT_APPLICATION_max	AMT_APPLICATION_mean
0	24835.5	24835.5	24835.500000
1	179055.0	179055.0	179055.000000
2	68809.5	900000.0	435436.500000
3	24282.0	24282.0	24282.000000
4	0.0	44617.5	22308.750000
5	0.0	688500.0	272203.260000
6	17176.5	247500.0	150530.250000
7	0.0	450000.0	155701.800000
8	40455.0	110160.0	76741.714286
9	247212.0	247212.0	247212.000000

	range_AMT_APPLICATION

```
0                   0.0
1                   0.0
2              831190.5
3                   0.0
4               44617.5
5              688500.0
6              230323.5
7              450000.0
8               69705.0
9                   0.0
```

```
agg_op_features
```

```
{'AMT_ANNUITY': [('AMT_ANNUITY_min', 'min'),
  ('AMT_ANNUITY_max', 'max'),
  ('AMT_ANNUITY_mean', 'mean')],
 'AMT_APPLICATION': [('AMT_APPLICATION_min', 'min'),
  ('AMT_APPLICATION_max', 'max'),
  ('AMT_APPLICATION_mean', 'mean')]}
```

```
result.isna().sum()
```

```
SK_ID_CURR                0
AMT_ANNUITY_min         480
AMT_ANNUITY_max         480
AMT_ANNUITY_mean        480
AMT_APPLICATION_min       0
AMT_APPLICATION_max       0
AMT_APPLICATION_mean      0
range_AMT_APPLICATION     0
dtype: int64
```

Sample Feature Engineering | Using feature transformer...

```
from sklearn.pipeline import make_pipeline

class prevAppsFeaturesAggregater(BaseEstimator, TransformerMixin):
    def __init__(self, features=None, prevApp=1): # no *args or
**kargs
        self.prevApp=prevApp
        self.features = features
        self.agg_op_features = {}
        for f in features:
            self.agg_op_features[f]=[]
            self.agg_op_features[f].extend((f"{f}_{func}",func) for
func in ["min", "max", "mean"])

    def fit(self, X, y=None):
        return self

    def transform(self, X, y=None):
        ####################-- Python Debugging---
```

```
################################
#         from IPython.core.debugger
#         import Pdb as pdb
#         pdb().set_trace()
#         breakpoint dont forget to quit
        ################################################################
        result = X.groupby(["SK_ID_CURR"]).agg(self.agg_op_features)
        result.columns = result.columns.droplevel()
        result = result.reset_index(level=["SK_ID_CURR"])
        if self.prevApp:
            result['range_AMT_APPLICATION'] =
result['AMT_APPLICATION_max'] - result['AMT_APPLICATION_min']
        return result
        # todo ---
        # return dataframe with the join key "SK_ID_CURR"



def test_driver_prevAppsFeaturesAggregater(df, features):
    print("Executing the test driver.............")
    print(f"df.shape: {df.shape}\n")
    print(f"df[{features}][0:5]: \n")
    display(df[features].head(5))
    print("---- Testing with `make_pipeline`---------")
    test_pipeline =
make_pipeline(prevAppsFeaturesAggregater(features))
    return(test_pipeline.fit_transform(df))



# All features of previous applications .....
features = ['AMT_ANNUITY',
       'AMT_APPLICATION', 'AMT_CREDIT', 'AMT_DOWN_PAYMENT',
'AMT_GOODS_PRICE',
       'RATE_DOWN_PAYMENT', 'RATE_INTEREST_PRIMARY',
       'RATE_INTEREST_PRIVILEGED', 'DAYS_DECISION',
'NAME_PAYMENT_TYPE',
       'CNT_PAYMENT',
       'DAYS_FIRST_DRAWING', 'DAYS_FIRST_DUE',
'DAYS_LAST_DUE_1ST_VERSION',
       'DAYS_LAST_DUE', 'DAYS_TERMINATION']
# Features of interest.....
features = ['AMT_ANNUITY', 'AMT_APPLICATION']

res = test_driver_prevAppsFeaturesAggregater(appsDF, features)
print("\n\n----- Results ----------")
print(f"Test driver: \n")
display(res.head(10))
print(f"input[features][0:10]: \n")
```

```
display(appsDF.head(10))
# QUESTION, should we lower case df['OCCUPATION_TYPE'] as Sales
staff != 'Sales Staff'? (hint: YES)

Executing the test driver.............
df.shape: (1670214, 37)

df[['AMT_ANNUITY', 'AMT_APPLICATION']][0:5]:


   AMT_ANNUITY  AMT_APPLICATION
0     1730.430          17145.0
1    25188.615         607500.0
2    15060.735         112500.0
3    47041.335         450000.0
4    31924.395         337500.0

---- Testing with `make_pipeline`---------


----- Results ----------
Test driver:


   SK_ID_CURR  AMT_ANNUITY_min  AMT_ANNUITY_max  AMT_ANNUITY_mean  \
0      100001         3951.000         3951.000       3951.000000
1      100002         9251.775         9251.775       9251.775000
2      100003         6737.310        98356.995      56553.990000
3      100004         5357.250         5357.250       5357.250000
4      100005         4813.200         4813.200       4813.200000
5      100006         2482.920        39954.510      23651.175000
6      100007         1834.290        22678.785      12278.805000
7      100008         8019.090        25309.575      15839.696250
8      100009         7435.845        17341.605      10051.412143
9      100010        27463.410        27463.410      27463.410000

   AMT_APPLICATION_min  AMT_APPLICATION_max  AMT_APPLICATION_mean  \
0              24835.5              24835.5          24835.500000
1             179055.0             179055.0         179055.000000
2              68809.5             900000.0         435436.500000
3              24282.0              24282.0          24282.000000
4                  0.0              44617.5          22308.750000
5                  0.0             688500.0         272203.260000
6              17176.5             247500.0         150530.250000
7                  0.0             450000.0         155701.800000
8              40455.0             110160.0          76741.714286
9             247212.0             247212.0         247212.000000

   range_AMT_APPLICATION
0                    0.0
```

```
1                   0.0
2                831190.5
3                   0.0
4                44617.5
5                688500.0
6                230323.5
7                450000.0
8                69705.0
9                   0.0

input[features][0:10]:


   SK_ID_PREV  SK_ID_CURR NAME_CONTRACT_TYPE  AMT_ANNUITY
AMT_APPLICATION  \
0     2030495      271877     Consumer loans     1730.430
17145.0
1     2802425      108129         Cash loans    25188.615
607500.0
2     2523466      122040         Cash loans    15060.735
112500.0
3     2819243      176158         Cash loans    47041.335
450000.0
4     1784265      202054         Cash loans    31924.395
337500.0
5     1383531      199383         Cash loans    23703.930
315000.0
6     2315218      175704         Cash loans          NaN
0.0
7     1656711      296299         Cash loans          NaN
0.0
8     2367563      342292         Cash loans          NaN
0.0
9     2579447      334349         Cash loans          NaN
0.0

   AMT_CREDIT  AMT_DOWN_PAYMENT  AMT_GOODS_PRICE
WEEKDAY_APPR_PROCESS_START  \
0     17145.0               0.0          17145.0
SATURDAY
1    679671.0               NaN         607500.0
THURSDAY
2    136444.5               NaN         112500.0
TUESDAY
3    470790.0               NaN         450000.0
MONDAY
4    404055.0               NaN         337500.0
THURSDAY
5    340573.5               NaN         315000.0
SATURDAY
```

```
6         0.0               NaN              NaN
TUESDAY
7         0.0               NaN              NaN
MONDAY
8         0.0               NaN              NaN
MONDAY
9         0.0               NaN              NaN
SATURDAY

   HOUR_APPR_PROCESS_START  ... NAME_SELLER_INDUSTRY  CNT_PAYMENT  \
0                       15  ...         Connectivity         12.0
1                       11  ...                  XNA         36.0
2                       11  ...                  XNA         12.0
3                        7  ...                  XNA         12.0
4                        9  ...                  XNA         24.0
5                        8  ...                  XNA         18.0
6                       11  ...                  XNA          NaN
7                        7  ...                  XNA          NaN
8                       15  ...                  XNA          NaN
9                       15  ...                  XNA          NaN

   NAME_YIELD_GROUP       PRODUCT_COMBINATION  DAYS_FIRST_DRAWING  \
0            middle  POS mobile with interest            365243.0
1        low_action          Cash X-Sell: low            365243.0
2              high         Cash X-Sell: high            365243.0
3            middle       Cash X-Sell: middle            365243.0
4              high         Cash Street: high                 NaN
5        low_normal          Cash X-Sell: low            365243.0
6               XNA                     Cash                 NaN
7               XNA                     Cash                 NaN
8               XNA                     Cash                 NaN
9               XNA                     Cash                 NaN

  DAYS_FIRST_DUE DAYS_LAST_DUE_1ST_VERSION  DAYS_LAST_DUE
DAYS_TERMINATION  \
0          -42.0                     300.0          -42.0            -
37.0
1         -134.0                     916.0       365243.0
365243.0
2         -271.0                      59.0       365243.0
365243.0
3         -482.0                    -152.0         -182.0           -
177.0
4            NaN                       NaN            NaN
NaN
5         -654.0                    -144.0         -144.0           -
137.0
6            NaN                       NaN            NaN
NaN
7            NaN                       NaN            NaN
```

```
NaN
8            NaN                         NaN            NaN
NaN
9            NaN                         NaN            NaN
NaN

  NFLAG_INSURED_ON_APPROVAL
0                       0.0
1                       1.0
2                       1.0
3                       1.0
4                       NaN
5                       1.0
6                       NaN
7                       NaN
8                       NaN
9                       NaN

[10 rows x 37 columns]
```

Join the labeled dataset

```
~3==3

False

datasets.keys()

dict_keys(['application_train', 'application_test', 'bureau',
'bureau_balance', 'credit_card_balance', 'installments_payments',
'previous_application', 'POS_CASH_balance'])

features = ['AMT_ANNUITY', 'AMT_APPLICATION']
bureau_features = ['AMT_ANNUITY', 'AMT_CREDIT_SUM']
bb_features = ['MONTHS_BALANCE']
ccb_features = ['MONTHS_BALANCE', 'AMT_BALANCE',
'CNT_INSTALMENT_MATURE_CUM']
ip_features = ['AMT_INSTALMENT', 'AMT_PAYMENT']

prevApps_feature_pipeline = Pipeline([
        # ('prevApps_add_features1', prevApps_add_features1()),  # add
some new features
        # ('prevApps_add_features2', prevApps_add_features2()),  # add
some new features
        ('prevApps_aggregater', prevAppsFeaturesAggregater(features)),
# Aggregate across old and new features
    ])
bureau_feature_pipeline = Pipeline([
        # ('prevApps_add_features1', prevApps_add_features1()),  # add
some new features
        # ('prevApps_add_features2', prevApps_add_features2()),  # add
some new features
```

```
        ('feature_aggregater',
prevAppsFeaturesAggregater(bureau_features,prevApp=0)), # Aggregate
across old and new features
    ])
bb_feature_pipeline = Pipeline([
        # ('prevApps_add_features1', prevApps_add_features1()),  # add
some new features
        # ('prevApps_add_features2', prevApps_add_features2()),  # add
some new features
        ('feature_aggregater',
prevAppsFeaturesAggregater(bb_features,prevApp=0)), # Aggregate across
old and new features
    ])
ccb_feature_pipeline = Pipeline([
        # ('prevApps_add_features1', prevApps_add_features1()),  # add
some new features
        # ('prevApps_add_features2', prevApps_add_features2()),  # add
some new features
        ('feature_aggregater',
prevAppsFeaturesAggregater(ccb_features,prevApp=0)), # Aggregate
across old and new features
    ])
ip_feature_pipeline = Pipeline([
        # ('prevApps_add_features1', prevApps_add_features1()),  # add
some new features
        # ('prevApps_add_features2', prevApps_add_features2()),  # add
some new features
        ('feature_aggregater',
prevAppsFeaturesAggregater(ip_features,prevApp=0)), # Aggregate across
old and new features
    ])
X_train= datasets["application_train"] #primary dataset
appsDF = datasets["previous_application"] #prev app


merge_all_data = True

# transform all the secondary tables
# 'bureau', 'bureau_balance', 'credit_card_balance',
'installments_payments',
# 'previous_application', 'POS_CASH_balance'

bureauDF = datasets['bureau']
bbDF = datasets['bureau_balance']
ccbDF = datasets['credit_card_balance']
ipDF = datasets['installments_payments']
posDF = datasets['POS_CASH_balance']

if merge_all_data:
    prevApps_aggregated = prevApps_feature_pipeline.transform(appsDF)
```

```
    bureau_aggregated = bureau_feature_pipeline.transform(bureauDF)
    # bb_aggregated = bb_feature_pipeline.transform(bbDF)
    ccb_aggregated = ccb_feature_pipeline.transform(ccbDF)
    ip_aggregated = ip_feature_pipeline.transform(ipDF)
    # pos_aggregated = prevApps_feature_pipeline.transform(posDF)


    #'bureau', 'bureau_balance', 'credit_card_balance',
'installments_payments',
    # 'previous_application', 'POS_CASH_balance'

# merge primary table and secondary tables using features based on
meta data and  aggregage stats
if merge_all_data:
    # 1. Join/Merge in prevApps Data
    X_train = X_train.merge(prevApps_aggregated, how='left',
on='SK_ID_CURR')

    # 2. Join/Merge in ...... Data
    X_train = X_train.merge(bureau_aggregated, how='left',
on="SK_ID_CURR")

    # 3. Join/Merge in .....Data
    dX_train = X_train.merge(ccb_aggregated, how='left',
on="SK_ID_CURR")

    # 4. Join/Merge in Aggregated ...... Data
    X_train = X_train.merge(ip_aggregated, how='left',
on="SK_ID_CURR")


print(X_train.shape)
display(X_train)
```

```
(307511, 141)
```

	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
...	...	...	...	...	...

```
307506      456251       0         Cash loans           M            N

307507      456252       0         Cash loans           F            N

307508      456253       0         Cash loans           F            N

307509      456254       1         Cash loans           F            N

307510      456255       0         Cash loans           F            N


       FLAG_OWN_REALTY  CNT_CHILDREN  AMT_INCOME_TOTAL  AMT_CREDIT  \
0                    Y             0          202500.0    406597.5
1                    N             0          270000.0   1293502.5
2                    Y             0           67500.0    135000.0
3                    Y             0          135000.0    312682.5
4                    Y             0          121500.0    513000.0
...                ...           ...               ...         ...
307506               N             0          157500.0    254700.0
307507               Y             0           72000.0    269550.0
307508               Y             0          153000.0    677664.0
307509               Y             0          171000.0    370107.0
307510               N             0          157500.0    675000.0

        AMT_ANNUITY  ...  AMT_ANNUITY_mean_y AMT_CREDIT_SUM_min  \
0           24700.5  ...                 0.0               0.0
1           35698.5  ...                 NaN           22248.0
2            6750.0  ...                 NaN           94500.0
3           29686.5  ...                 NaN               NaN
4           21865.5  ...                 NaN          146250.0
...             ...  ...                 ...               ...
307506      27558.0  ...                 NaN               NaN
307507      12001.5  ...                 NaN               NaN
307508      29979.0  ...             58369.5          360000.0
307509      20205.0  ...                 0.0           45000.0
307510      49117.5  ...              1081.5           22995.0

       AMT_CREDIT_SUM_max AMT_CREDIT_SUM_mean AMT_INSTALMENT_min  \
0               450000.0       108131.945625           9251.775
1               810000.0       254350.125000           6662.970
2                94537.8        94518.900000           5357.250
3                    NaN                 NaN           2482.920
4               146250.0       146250.000000           1821.780
...                  ...                 ...                ...
307506               NaN                 NaN           6605.910
307507               NaN                 NaN          10046.880
307508         2250000.0       990000.000000           2754.450
307509           45000.0        45000.000000           2296.440
```

```
307510           900000.0       345629.045455          11090.835

       AMT_INSTALMENT_max  AMT_INSTALMENT_mean  AMT_PAYMENT_min  \
0                53093.745         11559.247105         9251.775
1               560835.360         64754.586000         6662.970
2                10573.965          7096.155000         5357.250
3               691786.890         62947.088438         2482.920
4                22678.785         12666.444545            0.180
...                    ...                  ...              ...
307506           12815.010          7492.924286         6605.910
307507           10074.465         10069.867500        10046.880
307508            5575.185          4399.707857           27.270
307509           19065.825         10239.832895         2296.440
307510          615229.515         41464.713649           34.965

        AMT_PAYMENT_max  AMT_PAYMENT_mean
0             53093.745      11559.247105
1            560835.360      64754.586000
2             10573.965       7096.155000
3            691786.890      62947.088438
4             22678.785      12214.060227
...                 ...               ...
307506        12815.010       7492.924286
307507        10074.465      10069.867500
307508         5575.185       4115.915357
307509        19065.825      10239.832895
307510       669251.655      47646.215878

[307511 rows x 141 columns]
```

Test Data Feature Engineering

Join the unlabeled dataset (i.e., the submission file)

```
X_kaggle_test= datasets["application_test"]
if merge_all_data:
    # 1. Join/Merge in prevApps Data
    X_kaggle_test = X_kaggle_test.merge(prevApps_aggregated,
how='left', on='SK_ID_CURR')

    # 2. Join/Merge in ..... Data
    X_kaggle_test = X_kaggle_test.merge(bureau_aggregated, how='left',
on='SK_ID_CURR')

    # 3. Join/Merge in .....Data
    X_kaggle_test = X_kaggle_test.merge(ccb_aggregated, how='left',
on='SK_ID_CURR')

    # 4. Join/Merge in Aggregated ..... Data
```

```
    X_kaggle_test = X_kaggle_test.merge(ip_aggregated, how='left',
on='SK_ID_CURR')

print(X_kaggle_test.shape)
display(X_kaggle_test)
```

```
(48744, 149)
```

```
       SK_ID_CURR NAME_CONTRACT_TYPE CODE_GENDER FLAG_OWN_CAR
FLAG_OWN_REALTY  \
0          100001         Cash loans           F            N
Y
1          100005         Cash loans           M            N
Y
2          100013         Cash loans           M            Y
Y
3          100028         Cash loans           F            N
Y
4          100038         Cash loans           M            Y
N
...           ...                ...         ...          ...
...
48739      456221         Cash loans           F            N
Y
48740      456222         Cash loans           F            N
N
48741      456223         Cash loans           F            Y
Y
48742      456224         Cash loans           M            N
N
48743      456250         Cash loans           F            Y
N

       CNT_CHILDREN  AMT_INCOME_TOTAL  AMT_CREDIT  AMT_ANNUITY  \
0                 0          135000.0    568800.0      20560.5
1                 0           99000.0    222768.0      17370.0
2                 0          202500.0    663264.0      69777.0
3                 2          315000.0   1575000.0      49018.5
4                 1          180000.0    625500.0      32067.0
...             ...               ...         ...          ...
48739             0          121500.0    412560.0      17473.5
48740             2          157500.0    622413.0      31909.5
48741             1          202500.0    315000.0      33205.5
48742             0          225000.0    450000.0      25128.0
48743             0          135000.0    312768.0      24709.5

       AMT_GOODS_PRICE  ... AMT_BALANCE_mean
CNT_INSTALMENT_MATURE_CUM_min  \
0             450000.0  ...              NaN
NaN
```

```
1             180000.0  ...              NaN
NaN
2             630000.0  ...     18159.919219
1.0
3            1575000.0  ...      8085.058163
1.0
4             625500.0  ...              NaN
NaN
...                ...  ...              ...
...
48739         270000.0  ...              NaN
NaN
48740         495000.0  ...              NaN
NaN
48741         315000.0  ...              NaN
NaN
48742         450000.0  ...              NaN
NaN
48743         270000.0  ...    173589.326250
0.0

      CNT_INSTALMENT_MATURE_CUM_max CNT_INSTALMENT_MATURE_CUM_mean  \
0                               NaN                            NaN
1                               NaN                            NaN
2                              22.0                      18.719101
3                              35.0                      19.547619
4                               NaN                            NaN
...                             ...                            ...
48739                           NaN                            NaN
48740                           NaN                            NaN
48741                           NaN                            NaN
48742                           NaN                            NaN
48743                          10.0                       4.583333

      AMT_INSTALMENT_min  AMT_INSTALMENT_max  AMT_INSTALMENT_mean  \
0                3951.000           17397.900          5885.132143
1                4813.200           17656.245          6240.205000
2                  67.500          357347.745         10897.898516
3                   1.170           38988.540          4979.282257
4               11097.450           11100.600         11100.337500
...                   ...                 ...                  ...
48739           14222.430          244664.505         91036.455000
48740            3653.955           14571.765          8086.162192
48741           12640.950           81184.005         23158.991250
48742            5519.925           23451.705         17269.234138
48743               1.080           26474.625         13238.063100

       AMT_PAYMENT_min  AMT_PAYMENT_max  AMT_PAYMENT_mean
0            3951.000        17397.900       5885.132143
1            4813.200        17656.245       6240.205000
```

```
2                6.165       357347.745        9740.235774
3                1.170        38988.540        4356.731549
4            11097.450        11100.600       11100.337500
...                ...              ...               ...
48739        14222.430       244664.505       91036.455000
48740            2.700        14571.765        7771.447603
48741        12640.950        81184.005       23158.991250
48742         5519.925        23451.705       17269.234138
48743            1.080        26474.625       13044.983400

[48744 rows x 149 columns]
```

```
# approval rate 'NFLAG_INSURED_ON_APPROVAL'
```

Convert categorical features to numerical approximations (via pipeline)

```
# Convert categorical features to numerical approximations (via
pipeline)
class ClaimAttributesAdder(BaseEstimator, TransformerMixin):
    def fit(self, X, y=None):
        return self
    def transform(self, X, y=None):
        charlson_idx_dt = {'0': 0, '1-2': 2, '3-4': 4, '5+': 6}
        los_dt = {'1 day': 1, '2 days': 2, '3 days': 3, '4 days': 4,
'5 days': 5, '6 days': 6,
          '1- 2 weeks': 11, '2- 4 weeks': 21, '4- 8 weeks': 42, '26+
weeks': 180}
        X['PayDelay'] = X['PayDelay'].apply(lambda x: int(x) if x !=
'162+' else int(162))
        X['DSFS'] = X['DSFS'].apply(lambda x: None if pd.isnull(x)
else int(x[0]) + 1)
        X['CharlsonIndex'] = X['CharlsonIndex'].apply(lambda x:
charlson_idx_dt[x])
        X['LengthOfStay'] = X['LengthOfStay'].apply(lambda x: None if
pd.isnull(x) else los_dt[x])
        return X
```

Sample Processing pipeline | Known Issues

OHE when previously unseen unique values in the test/validation set

Train, validation and Test sets (and the leakage problem we have mentioned previously):

Let's look at a small usecase to tell us how to deal with this:

- The OneHotEncoder is fitted to the training set, which means that for each unique value present in the training set, for each feature, a new column is created. Let's say we have 39 columns after the encoding up from 30 (before preprocessing).

- The output is a numpy array (when the option sparse=False is used), which has the disadvantage of losing all the information about the original column names and values.
- When we try to transform the test set, after having fitted the encoder to the training set, we obtain a `ValueError`. This is because the there are new, previously unseen unique values in the test set and the encoder doesn't know how to handle these values. In order to use both the transformed training and test sets in machine learning algorithms, we need them to have the same number of columns.

This last problem can be solved by using the option handle_unknown='ignore'of the OneHotEncoder, which, as the name suggests, will ignore previously unseen values when transforming the test set.

Here is a example that in action:

```
# Identify the categorical features we wish to consider.
cat_attribs = ['CODE_GENDER',
'FLAG_OWN_REALTY','FLAG_OWN_CAR','NAME_CONTRACT_TYPE',

'NAME_EDUCATION_TYPE','OCCUPATION_TYPE','NAME_INCOME_TYPE']

# Notice handle_unknown="ignore" in OHE which ignore values from the
validation/test that
# do NOT occur in the training set
cat_pipeline = Pipeline([
        ('selector', DataFrameSelector(cat_attribs)),
        ('imputer', SimpleImputer(strategy='most_frequent')),
        ('ohe', OneHotEncoder(sparse=False, handle_unknown="ignore"))
    ])
```

OHE case study: The breast cancer wisconsin dataset (classification)

```
from sklearn.datasets import load_breast_cancer
data = load_breast_cancer(return_X_y=False)
X, y = load_breast_cancer(return_X_y=True)
print(y[[10, 50, 85]])
#([0, 1, 0])
list(data.target_names)
#['malignant', 'benign']
X.shape
```

```
[0 1 0]
```

```
(569, 30)
```

```
data.feature_names
```

```
array(['mean radius', 'mean texture', 'mean perimeter', 'mean area',
       'mean smoothness', 'mean compactness', 'mean concavity',
       'mean concave points', 'mean symmetry', 'mean fractal
dimension',
```

```
       'radius error', 'texture error', 'perimeter error', 'area
error',
       'smoothness error', 'compactness error', 'concavity error',
       'concave points error', 'symmetry error',
       'fractal dimension error', 'worst radius', 'worst texture',
       'worst perimeter', 'worst area', 'worst smoothness',
       'worst compactness', 'worst concavity', 'worst concave points',
       'worst symmetry', 'worst fractal dimension'], dtype='<U23')
```

Please this blog for more details of OHE when the validation/test have previously unseen unique values.

HCDR preprocessing

Feature Engineering | Feature Selections | Preprocessing

```
from sklearn.model_selection import train_test_split
```

Train, Valid, Test dataset selection

`use_application_data_ONLY`: True : use the application data , False use feature Engineered data

```
# `use_application_data_ONLY`: True : use the application data , False
use feature Engineered data
use_application_data_ONLY = True

# Split the provided training data into training and validationa and
test
# The kaggle evaluation test set has no labels

def load_train_valid_test_data(use_application_data_ONLY=False):
    global X_train, X_valid, X_test, y_train, y_valid, y_test,
selected_features

    if use_application_data_ONLY:
        print("use_application_data_ONLY? :
{}".format(use_application_data_ONLY))
        print("Baseline Model.......\n\n")
        print("Loding just application data.......\n\n")

        # just selected a few features for a baseline experiment
        selected_features = ['AMT_INCOME_TOTAL',
'AMT_CREDIT','DAYS_EMPLOYED','DAYS_BIRTH','EXT_SOURCE_1',
            'EXT_SOURCE_2','EXT_SOURCE_3','CODE_GENDER',
'FLAG_OWN_REALTY','FLAG_OWN_CAR','NAME_CONTRACT_TYPE',

'NAME_EDUCATION_TYPE','OCCUPATION_TYPE','NAME_INCOME_TYPE']

        X_train = datasets["application_train"][selected_features]
        y_train = datasets["application_train"]['TARGET']
```

```
        X_train, X_valid, y_train, y_valid = train_test_split(X_train,
y_train, test_size=0.15, random_state=42)

        X_train, X_test, y_train, y_test = train_test_split(X_train,
y_train, test_size=0.15, random_state=42)

        X_kaggle_test= datasets["application_test"][selected_features]

        # y_test = datasets["application_test"]['TARGET']   #why no
TARGET?!! (hint: kaggle competition)
    else:
        selected_features = ['AMT_INCOME_TOTAL',
'AMT_CREDIT','DAYS_EMPLOYED','DAYS_BIRTH','EXT_SOURCE_1',
            'EXT_SOURCE_2','EXT_SOURCE_3','CODE_GENDER',
'FLAG_OWN_REALTY','FLAG_OWN_CAR','NAME_CONTRACT_TYPE',

'NAME_EDUCATION_TYPE','OCCUPATION_TYPE','NAME_INCOME_TYPE']
        X_train = datasets["application_train"]
        y_train = X_train['TARGET']
        X_train = X_train[selected_features]

        X_train, X_valid, y_train, y_valid = train_test_split(X_train,
y_train, test_size=0.15, random_state=42)

        X_train, X_test, y_train, y_test = train_test_split(X_train,
y_train, test_size=0.15, random_state=42)
        X_kaggle_test= datasets["application_test"][selected_features]

        X_kaggle_test= X_kaggle_test[selected_features]
        # y_test = datasets["application_test"]['TARGET']   #why no
TARGET?!! (hint: kaggle competition)

    print("-------------------------------------------------")
    print(f"X train           shape: {X_train.shape}")
    print(f"X validation      shape: {X_valid.shape}")
    print(f"X test            shape: {X_test.shape}")
    print(f"X X_kaggle_test   shape: {X_kaggle_test.shape}")
    print(f"Y train           shape: {y_train.shape}")
    print(f"Y validation      shape: {y_valid.shape}")
    print(f"Y test            shape: {y_test.shape}")

load_train_valid_test_data(False)
```

```
-------------------------------------------------
X train           shape: (222176, 14)
X validation      shape: (46127, 14)
X test            shape: (39208, 14)
X X_kaggle_test   shape: (48744, 14)
Y train           shape: (222176,)
```

```
Y validation      shape: (46127,)
Y test            shape: (39208,)
```

Feature Enginnering

Experimental Features

Todo's

- ☐ Data Aggregation
  - ☐ `Pipelines` for all secondary tables
  - ☐ For table `bureau`
  - ☐ For table `previous_applications`
- ☐ Merge the aggregated data in Main table using Pipeline
- ☐ Include Imputation, Scaling, Normalizing
- ☐ Define the Numerical And cateogrical Features
  - ☐ use `Feature Union` to combine multiple Pipelines
- ☐ Combine above pipeline with the Predicator Pipeline
  - ☐ Parameter Hypertuning

```
s_data = bur[bur["SK_ID_CURR"]==100001 ]
display(s_data)
```

```
        SK_ID_CURR  SK_ID_BUREAU CREDIT_ACTIVE CREDIT_CURRENCY
DAYS_CREDIT  \
248484      100001       5896630        Closed      currency 1
-857
248485      100001       5896631        Closed      currency 1
-909
248486      100001       5896632        Closed      currency 1
-879
248487      100001       5896633        Closed      currency 1
-1572
248488      100001       5896634        Active      currency 1
-559
248489      100001       5896635        Active      currency 1
-49
248490      100001       5896636        Active      currency 1
-320

        CREDIT_DAY_OVERDUE  DAYS_CREDIT_ENDDATE  DAYS_ENDDATE_FACT  \
248484                   0               -492.0             -553.0
248485                   0               -179.0             -877.0
248486                   0               -514.0             -544.0
248487                   0              -1329.0            -1328.0
248488                   0                902.0                NaN
248489                   0               1778.0                NaN
```

```
248490                   0                411.0                 NaN

        AMT_CREDIT_MAX_OVERDUE  CNT_CREDIT_PROLONG  AMT_CREDIT_SUM  \
248484                     NaN                   0        112500.0
248485                     NaN                   0        279720.0
248486                     NaN                   0         91620.0
248487                     NaN                   0         85500.0
248488                     NaN                   0        337680.0
248489                     NaN                   0        378000.0
248490                     NaN                   0        168345.0

        AMT_CREDIT_SUM_DEBT  AMT_CREDIT_SUM_LIMIT
AMT_CREDIT_SUM_OVERDUE  \
248484                  0.0                   0.0
0.0
248485                  0.0                   0.0
0.0
248486                  0.0                   0.0
0.0
248487                  0.0                   0.0
0.0
248488             113166.0                   0.0
0.0
248489             373239.0                   0.0
0.0
248490             110281.5                   NaN
0.0

            CREDIT_TYPE  DAYS_CREDIT_UPDATE  AMT_ANNUITY
248484  Consumer credit                -155          0.0
248485  Consumer credit                -155          0.0
248486  Consumer credit                -155          0.0
248487  Consumer credit                -155          0.0
248488  Consumer credit                  -6       4630.5
248489  Consumer credit                 -16      10822.5
248490  Consumer credit                 -10       9364.5
```

```
# Total number of Past loans ?
s_data[["SK_ID_CURR",
"CREDIT_ACTIVE"]].groupby("SK_ID_CURR").count().reset_index().rename(c
olumns = {'CREDIT_ACTIVE':'TOTAL_PAST_LOANS'})
# left join with main table
```

```
   SK_ID_CURR  TOTAL_PAST_LOANS
0      100001                 7
```

```
# Total types of loan
s_data[["SK_ID_CURR",
"CREDIT_TYPE"]].groupby("SK_ID_CURR").nunique().reset_index().rename(c
```

```
olumns = {'CREDIT_TYPE':'TOTAL_TYPES_OF_LOAN'})
# left join with main table
```

```
   SK_ID_CURR  TOTAL_TYPES_OF_LOAN
0      100001                   1
```

```
# % of active loans
s_data["is_credit_active"]= s_data[["CREDIT_ACTIVE"]].apply(func=
lambda x: False if x.CREDIT_ACTIVE=="Closed" else True, axis=1)
s_data[["SK_ID_CURR",
"is_credit_active"]].groupby("SK_ID_CURR").mean().reset_index().rename
(columns = {'is_credit_active':'ACTIVE_LOANS_MEAN'})
# left join with main table
```

```
   SK_ID_CURR  ACTIVE_LOANS_MEAN
0      100001           0.428571
```

```
# average of (days to credit end) for active credit.


with_active_credits = s_data[s_data["is_credit_active"]]
# display(with_active_credits)
if len(with_active_credits):
  print(with_active_credits[["SK_ID_CURR",
"DAYS_CREDIT_ENDDATE"]].groupby("SK_ID_CURR").mean().reset_index().ren
ame(columns={'DAYS_CREDIT_ENDDATE':
'DAYS_CREDIT_ENDDATE_NORMALIZED_MEAN'}))
  # left join with main table

# ! WARNING: When joining above dataframe with main table, do not
forget to fill
# empty values in column `DAYS_CREDIT_ENDDATE_NORMALIZED_MEAN` with 0
since
# there will be some applicants who will have no active credits thus
`DAYS_CREDIT_ENDDATE` will be np.NaN( after joining)
# so you don't want to impute the nan values in this column
```

```
   SK_ID_CURR  DAYS_CREDIT_ENDDATE_NORMALIZED_MEAN
0      100001                          1030.333333
```

```
# mean number of prolonged credits

# s_data[~s_data["CNT_CREDIT_PROLONG"].isna()]

# mean overdue loands with % of active loans ??????
bur[~bur["AMT_CREDIT_MAX_OVERDUE"].isna()][bur.SK_ID_CURR == 215354]

# % of utilized debt??
```

```
   SK_ID_CURR  SK_ID_BUREAU CREDIT_ACTIVE CREDIT_CURRENCY  DAYS_CREDIT
\
4      215354       5714466        Active      currency 1         -629

5      215354       5714467        Active      currency 1         -273

6      215354       5714468        Active      currency 1          -43


   CREDIT_DAY_OVERDUE  DAYS_CREDIT_ENDDATE  DAYS_ENDDATE_FACT  \
4                   0               1197.0                NaN
5                   0              27460.0                NaN
6                   0                 79.0                NaN

   AMT_CREDIT_MAX_OVERDUE  CNT_CREDIT_PROLONG  AMT_CREDIT_SUM  \
4                 77674.5                   0       2700000.0
5                     0.0                   0        180000.0
6                     0.0                   0         42103.8

   AMT_CREDIT_SUM_DEBT  AMT_CREDIT_SUM_LIMIT
AMT_CREDIT_SUM_OVERDUE  \
4                  NaN                   NaN                      0.0

5             71017.38             108982.62                      0.0

6             42103.80                  0.00                      0.0


       CREDIT_TYPE  DAYS_CREDIT_UPDATE  AMT_ANNUITY
4  Consumer credit                 -21          NaN
5      Credit card                 -31          NaN
6  Consumer credit                 -22          NaN
```

```
class SecondaryTableAgg(BaseEstimator, TransformerMixin):
    def __init__(self, table_name, features=None, agg_features=None,
prevApp=1):
        self.data = datasets["table_name"]
        self.features = features
        # Todo - use mix of features from `agg_features`
        self.agg_op_features = {}
        for f in features:
            self.agg_op_features[f]=[]
            self.agg_op_features[f].extend((f"{f}_{func}",func) for
func in ["min", "max", "mean"])
        print("Inside the `SecondaryTableAgg` Transformer....")

    def fit(self, X, y=None):
        return self
```

```
    def transform(self, X, y=None):
        ####################-- Python Debugging---
################################
#         from IPython.core.debugger
#         import Pdb as pdb
#         pdb().set_trace()
#         breakpoint dont forget to quit
        ###############################################################
        result = X.groupby(["SK_ID_CURR"]).agg(self.agg_op_features)
        result.columns = result.columns.droplevel()
        result = result.reset_index(level=["SK_ID_CURR"])
        if self.prevApp:
            result['range_AMT_APPLICATION'] =
result['AMT_APPLICATION_max'] - result['AMT_APPLICATION_min']
        return result
        # todo ---
        # return dataframe with the join key "SK_ID_CURR"

features = ['AMT_ANNUITY', 'AMT_APPLICATION']
bureau_features = ['AMT_ANNUITY', 'AMT_CREDIT_SUM']
bb_features = ['MONTHS_BALANCE']
ccb_features = ['MONTHS_BALANCE', 'AMT_BALANCE',
'CNT_INSTALMENT_MATURE_CUM']
ip_features = ['AMT_INSTALMENT', 'AMT_PAYMENT']

prevApps_feature_pipeline = Pipeline([
        ('prevApps_aggregater', prevAppsFeaturesAggregater(features)),
# Aggregate across old and new features
    ])
bureau_feature_pipeline = Pipeline([
        ('feature_aggregater',
prevAppsFeaturesAggregater(bureau_features,prevApp=0)), # Aggregate
across old and new features
    ])
bb_feature_pipeline = Pipeline([
        # ('prevApps_add_features1', prevApps_add_features1()),  # add
some new features
        # ('prevApps_add_features2', prevApps_add_features2()),  # add
some new features
        ('feature_aggregater',
prevAppsFeaturesAggregater(bb_features,prevApp=0)), # Aggregate across
old and new features
    ])
ccb_feature_pipeline = Pipeline([
        # ('prevApps_add_features1', prevApps_add_features1()),  # add
some new features
        # ('prevApps_add_features2', prevApps_add_features2()),  # add
some new features
        ('feature_aggregater',
prevAppsFeaturesAggregater(ccb_features,prevApp=0)), # Aggregate
```

```
across old and new features
    ])
ip_feature_pipeline = Pipeline([
        # ('prevApps_add_features1', prevApps_add_features1()),  # add
some new features
        # ('prevApps_add_features2', prevApps_add_features2()),  # add
some new features
        ('feature_aggregater',
prevAppsFeaturesAggregater(ip_features,prevApp=0)), # Aggregate across
old and new features
    ])
X_train= datasets["application_train"] #primary dataset
appsDF = datasets["previous_application"] #prev app


merge_all_data = True

# transform all the secondary tables
# 'bureau', 'bureau_balance', 'credit_card_balance',
'installments_payments',
# 'previous_application', 'POS_CASH_balance'

bureauDF = datasets['bureau']
bbDF = datasets['bureau_balance']
ccbDF = datasets['credit_card_balance']
ipDF = datasets['installments_payments']
posDF = datasets['POS_CASH_balance']

if merge_all_data:
    prevApps_aggregated = prevApps_feature_pipeline.transform(appsDF)
    bureau_aggregated = bureau_feature_pipeline.transform(bureauDF)
    # bb_aggregated = bb_feature_pipeline.transform(bbDF)
    ccb_aggregated = ccb_feature_pipeline.transform(ccbDF)
    ip_aggregated = ip_feature_pipeline.transform(ipDF)
    # pos_aggregated = prevApps_feature_pipeline.transform(posDF)


    #'bureau', 'bureau_balance', 'credit_card_balance',
'installments_payments',
    # 'previous_application', 'POS_CASH_balance'

# merge primary table and secondary tables using features based on
meta data and  aggregage stats
if merge_all_data:
    # 1. Join/Merge in prevApps Data
    X_train = X_train.merge(prevApps_aggregated, how='left',
on='SK_ID_CURR')

    # 2. Join/Merge in ...... Data
```

```
    X_train = X_train.merge(bureau_aggregated, how='left',
on="SK_ID_CURR")

    # 3. Join/Merge in .....Data
    dX_train = X_train.merge(ccb_aggregated, how='left',
on="SK_ID_CURR")

    # 4. Join/Merge in Aggregated ...... Data
    X_train = X_train.merge(ip_aggregated, how='left',
on="SK_ID_CURR")


print(X_train.shape)
display(X_train)

class prevAppsFeaturesAggregater(BaseEstimator, TransformerMixin):
    def __init__(self, features=None, prevApp=1): # no *args or
**kargs
        self.prevApp=prevApp
        self.features = features
        self.agg_op_features = {}
        for f in features:
            self.agg_op_features[f]=[]
            self.agg_op_features[f].extend((f"{f}_{func}",func) for
func in ["min", "max", "mean"])

    def fit(self, X, y=None):
        return self

    def transform(self, X, y=None):
        ######################-- Python Debugging---
################################
#         from IPython.core.debugger
#         import Pdb as pdb
#         pdb().set_trace()
#         breakpoint dont forget to quit
        ################################################################
        result = X.groupby(["SK_ID_CURR"]).agg(self.agg_op_features)
        result.columns = result.columns.droplevel()
        result = result.reset_index(level=["SK_ID_CURR"])
        if self.prevApp:
            result['range_AMT_APPLICATION'] =
result['AMT_APPLICATION_max'] - result['AMT_APPLICATION_min']
        return result
        # todo ---
        # return dataframe with the join key "SK_ID_CURR"



def test_driver_prevAppsFeaturesAggregater(df, features):
```

```
    print("Executing the test driver.............")
    print(f"df.shape: {df.shape}\n")
    print(f"df[{features}][0:5]: \n")
    display(df[features].head(5))
    print("---- Testing with `make_pipeline`---------")
    test_pipeline =
make_pipeline(prevAppsFeaturesAggregater(features))
    return(test_pipeline.fit_transform(df))



# All features of previous applications .....
features = ['AMT_ANNUITY',
       'AMT_APPLICATION', 'AMT_CREDIT', 'AMT_DOWN_PAYMENT',
'AMT_GOODS_PRICE',
       'RATE_DOWN_PAYMENT', 'RATE_INTEREST_PRIMARY',
       'RATE_INTEREST_PRIVILEGED', 'DAYS_DECISION',
'NAME_PAYMENT_TYPE',
       'CNT_PAYMENT',
       'DAYS_FIRST_DRAWING', 'DAYS_FIRST_DUE',
'DAYS_LAST_DUE_1ST_VERSION',
       'DAYS_LAST_DUE', 'DAYS_TERMINATION']
# Features of interest.....
features = ['AMT_ANNUITY', 'AMT_APPLICATION']

res = test_driver_prevAppsFeaturesAggregater(appsDF, features)
print("\n\n----- Results ----------")
print(f"Test driver: \n")
display(res.head(10))
print(f"input[features][0:10]: \n")
display(appsDF.head(10))
# QUESTION, should we lower case df['OCCUPATION_TYPE'] as Sales
staff != 'Sales Staff'? (hint: YES)

class prevAppsFeaturesAggregater(BaseEstimator, TransformerMixin):
    def __init__(self, features=None, prevApp=1): # no *args or
**kargs
        self.prevApp=prevApp
        self.features = features
        self.agg_op_features = {}
        for f in features:
            self.agg_op_features[f]=[]
            self.agg_op_features[f].extend((f"{f}_{func}",func) for
func in ["min", "max", "mean"])

    def fit(self, X, y=None):
        return self

    def transform(self, X, y=None):
        ####################-- Python Debugging---
```

```
################################
#         from IPython.core.debugger
#         import Pdb as pdb
#         pdb().set_trace()
#         breakpoint dont forget to quit
        ################################################################
        result = X.groupby(["SK_ID_CURR"]).agg(self.agg_op_features)
        result.columns = result.columns.droplevel()
        result = result.reset_index(level=["SK_ID_CURR"])
        if self.prevApp:
            result['range_AMT_APPLICATION'] =
result['AMT_APPLICATION_max'] - result['AMT_APPLICATION_min']
        return result
        # todo ---
        # return dataframe with the join key "SK_ID_CURR"



def test_driver_prevAppsFeaturesAggregater(df, features):
    print("Executing the test driver.............")
    print(f"df.shape: {df.shape}\n")
    print(f"df[{features}][0:5]: \n")
    display(df[features].head(5))
    print("---- Testing with `make_pipeline`---------")
    test_pipeline =
make_pipeline(prevAppsFeaturesAggregater(features))
    return(test_pipeline.fit_transform(df))



# All features of previous applications .....
features = ['AMT_ANNUITY',
       'AMT_APPLICATION', 'AMT_CREDIT', 'AMT_DOWN_PAYMENT',
'AMT_GOODS_PRICE',
       'RATE_DOWN_PAYMENT', 'RATE_INTEREST_PRIMARY',
       'RATE_INTEREST_PRIVILEGED', 'DAYS_DECISION',
'NAME_PAYMENT_TYPE',
       'CNT_PAYMENT',
       'DAYS_FIRST_DRAWING', 'DAYS_FIRST_DUE',
'DAYS_LAST_DUE_1ST_VERSION',
       'DAYS_LAST_DUE', 'DAYS_TERMINATION']
# Features of interest.....
features = ['AMT_ANNUITY', 'AMT_APPLICATION']

res = test_driver_prevAppsFeaturesAggregater(appsDF, features)
print("\n\n----- Results ----------")
print(f"Test driver: \n")
display(res.head(10))
print(f"input[features][0:10]: \n")
```

```
display(appsDF.head(10))
# QUESTION, should we lower case df['OCCUPATION_TYPE'] as Sales
staff != 'Sales Staff'? (hint: YES)

from sklearn.base import BaseEstimator, TransformerMixin
import re

# Creates the following date features
# But could do so much more with these features
#    E.g.,
#      extract the domain address of the homepage and OneHotEncode it
#
# ['release_month','release_day','release_year',
'release_dayofweek','release_quarter']
class prep_OCCUPATION_TYPE(BaseEstimator, TransformerMixin):
    def __init__(self, features="OCCUPATION_TYPE"): # no *args or
**kargs
        self.features = features
    def fit(self, X, y=None):
        return self  # nothing else to do
    def transform(self, X):
        df = pd.DataFrame(X, columns=self.features)
        #from IPython.core.debugger import Pdb as pdb;
pdb().set_trace() #breakpoint; dont forget to quit
        df['OCCUPATION_TYPE'] = df['OCCUPATION_TYPE'].apply(lambda x:
1. if x in ['Core Staff', 'Accountants', 'Managers', 'Sales Staff',
'Medicine Staff', 'High Skill Tech Staff', 'Realty Agents', 'IT
Staff', 'HR Staff'] else 0.)
        #df.drop(self.features, axis=1, inplace=True)
        return np.array(df.values)  #return a Numpy Array to observe
the pipeline protocol


from sklearn.pipeline import make_pipeline
features = ["OCCUPATION_TYPE"]
def test_driver_prep_OCCUPATION_TYPE():
    print(f"X_train.shape: {X_train.shape}\n")
    print(f"X_train['name'][0:5]: \n{X_train[features][0:5]}")
    test_pipeline = make_pipeline(prep_OCCUPATION_TYPE(features))
    return(test_pipeline.fit_transform(X_train))

x = test_driver_prep_OCCUPATION_TYPE()
print(f"Test driver: \n{test_driver_prep_OCCUPATION_TYPE()[0:10, :]}")
print(f"X_train['name'][0:10]: \n{X_train[features][0:10]}")


# QUESTION, should we lower case df['OCCUPATION_TYPE'] as Sales
staff != 'Sales Staff'? (hint: YES)
```

```
# Convert categorical features to numerical approximations (via
pipeline)
class ClaimAttributesAdder(BaseEstimator, TransformerMixin):
    def fit(self, X, y=None):
        return self
    def transform(self, X, y=None):
        charlson_idx_dt = {'0': 0, '1-2': 2, '3-4': 4, '5+': 6}
        los_dt = {'1 day': 1, '2 days': 2, '3 days': 3, '4 days': 4,
'5 days': 5, '6 days': 6,
          '1- 2 weeks': 11, '2- 4 weeks': 21, '4- 8 weeks': 42, '26+
weeks': 180}
        X['PayDelay'] = X['PayDelay'].apply(lambda x: int(x) if x !=
'162+' else int(162))
        X['DSFS'] = X['DSFS'].apply(lambda x: None if pd.isnull(x)
else int(x[0]) + 1)
        X['CharlsonIndex'] = X['CharlsonIndex'].apply(lambda x:
charlson_idx_dt[x])
        X['LengthOfStay'] = X['LengthOfStay'].apply(lambda x: None if
pd.isnull(x) else los_dt[x])
        return X
```

Data Pipeline

```
from sklearn import set_config


# Create a class to select numerical or categorical columns
# since Scikit-Learn doesn't handle DataFrames yet
class DataFrameSelector(BaseEstimator, TransformerMixin):
    def __init__(self, attribute_names):
        self.attribute_names = attribute_names
    def fit(self, X, y=None):
        return self
    def transform(self, X):
        return X[self.attribute_names].values
```

```
# Identify the numeric features we wish to consider.
num_attribs = [
    'AMT_INCOME_TOTAL',
'AMT_CREDIT','DAYS_EMPLOYED','DAYS_BIRTH','EXT_SOURCE_1',
    'EXT_SOURCE_2','EXT_SOURCE_3']

num_pipeline = Pipeline([
        ('selector', DataFrameSelector(num_attribs)),
        ('imputer', SimpleImputer(strategy='mean')),
        ('std_scaler', StandardScaler()),
    ])
# Identify the categorical features we wish to consider.
cat_attribs = ['CODE_GENDER',
'FLAG_OWN_REALTY','FLAG_OWN_CAR','NAME_CONTRACT_TYPE',

'NAME_EDUCATION_TYPE','OCCUPATION_TYPE','NAME_INCOME_TYPE']

# Notice handle_unknown="ignore" in OHE which ignore values from the
validation/test that
# do NOT occur in the training set
cat_pipeline = Pipeline([
        ('selector', DataFrameSelector(cat_attribs)),
        #('imputer', SimpleImputer(strategy='most_frequent')),
        ('imputer', SimpleImputer(strategy='constant',
fill_value='missing')),
        ('ohe', OneHotEncoder(sparse=False, handle_unknown="ignore"))
    ])

data_prep_pipeline = FeatureUnion(transformer_list=[
        ("num_pipeline", num_pipeline),
        ("cat_pipeline", cat_pipeline),
    ])

set_config(display="diagram")
data_prep_pipeline
```

```
FeatureUnion(transformer_list=[('num_pipeline',
                                Pipeline(steps=[('selector',

DataFrameSelector(attribute_names=['AMT_INCOME_TOTAL',

'AMT_CREDIT',

'DAYS_EMPLOYED',

'DAYS_BIRTH',

'EXT_SOURCE_1',
```

```
'EXT_SOURCE_2',

'EXT_SOURCE_3'])),
                                                 ('imputer',
SimpleImputer()),
                                                 ('std_scaler',
                                                  StandardScaler())])),
                                ('cat_pipeline',
                                 Pipeline(steps=[('selector',

DataFrameSelector(attribute_names=['CODE_GENDER',

'FLAG_OWN_REALTY',

'FLAG_OWN_CAR',

'NAME_CONTRACT_TYPE',

'NAME_EDUCATION_TYPE',

'OCCUPATION_TYPE',

'NAME_INCOME_TYPE'])),
                                                 ('imputer',

SimpleImputer(fill_value='missing',

strategy='constant')),
                                                 ('ohe',

OneHotEncoder(handle_unknown='ignore',

sparse=False))]))])])
```

```
list(datasets["application_train"].columns)
```

```
['SK_ID_CURR',
 'TARGET',
 'NAME_CONTRACT_TYPE',
 'CODE_GENDER',
 'FLAG_OWN_CAR',
 'FLAG_OWN_REALTY',
 'CNT_CHILDREN',
 'AMT_INCOME_TOTAL',
 'AMT_CREDIT',
 'AMT_ANNUITY',
 'AMT_GOODS_PRICE',
 'NAME_TYPE_SUITE',
 'NAME_INCOME_TYPE',
```

```
'NAME_EDUCATION_TYPE',
'NAME_FAMILY_STATUS',
'NAME_HOUSING_TYPE',
'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',
'OCCUPATION_TYPE',
'CNT_FAM_MEMBERS',
'REGION_RATING_CLIENT',
'REGION_RATING_CLIENT_W_CITY',
'WEEKDAY_APPR_PROCESS_START',
'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',
'ORGANIZATION_TYPE',
'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',
'FONDKAPREMONT_MODE',
'HOUSETYPE_MODE',
'TOTALAREA_MODE',
'WALLSMATERIAL_MODE',
'EMERGENCYSTATE_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']
```

Modeling

```
def pct(x):
    return round(100*x,3)

expLog_columns = ["exp_name","Train Acc", "Valid Acc","Test
Acc","Train AUC", "Valid AUC","Test AUC"]
try:
    expLog
except NameError:
    expLog = pd.DataFrame(columns=expLog_columns)
    use_application_data_ONLY=True

load_train_valid_test_data()
```

```
--------------------------------------------------
X train           shape: (222176, 14)
X validation      shape: (46127, 14)
X test            shape: (39208, 14)
X X_kaggle_test   shape: (48744, 14)
Y train           shape: (222176,)
Y validation      shape: (46127,)
Y test            shape: (39208,)
```

Baseline Model

To get a baseline, we will use some of the features after being preprocessed through the pipeline. The baseline model is a logistic regression model

```
%%time
np.random.seed(42)

full_pipeline_with_predictor = Pipeline([
        ("preparation", data_prep_pipeline),
        ("linear", LogisticRegression())
    ])
model = full_pipeline_with_predictor.fit(X_train, y_train)
```

```
CPU times: user 6.06 s, sys: 2.22 s, total: 8.29 s
Wall time: 4.56 s
```

Gridsearch with CV

```
%%time
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import SGDClassifier
from sklearn.naive_bayes import BernoulliNB
from sklearn.metrics import accuracy_score, roc_auc_score
from sklearn import set_config

load_train_valid_test_data()

print("-----------------------------------------\n\n Grid
Search......\n\n")

full_pipeline_with_predictor = Pipeline([
        ("preparation", data_prep_pipeline),
        ("clf", LogisticRegression()),
    ])
# Todo- can we try different imputation and scaling in param grid??
param_grid = [

  {
      'clf': (LogisticRegression(),),
  },
  {
      'clf': (BernoulliNB(),),
  },
  {
      'clf': (SGDClassifier(),),
  },
]

gsv = GridSearchCV(
    full_pipeline_with_predictor, param_grid,
    cv=5, n_jobs=-1, verbose=3, return_train_score=True,
scoring="roc_auc"
)
model = gsv.fit(X_train, y_train)

print("The best roc_auc_score is: {}".format(model.best_score_))
print("------ The best parameters are: {}".format(model.best_params_))
print("The accuracy score of this model is:
{}".format(np.round(accuracy_score(y_train, model.predict(X_train)),
3)))
print("\n\nGrid search Results:-----")

required_cols = ["params", "rank_test_score", "mean_train_score",
"std_train_score", "mean_test_score", "std_test_score"]
display(pd.DataFrame(model.cv_results_)
[required_cols].sort_values(by="rank_test_score"))
```

```
print("-------------------------------------------------------")
print("Experiment results so far......")
exp_name = f"Gridserach_baseline{len(selected_features)}_features"
expLog.loc[len(expLog)] = [f"{exp_name}"] + list(np.round(
               [accuracy_score(y_train, model.predict(X_train)),
                accuracy_score(y_valid, model.predict(X_valid)),
                accuracy_score(y_test, model.predict(X_test)),
                roc_auc_score(y_train, model.predict_proba(X_train)[:,
1]),
                roc_auc_score(y_valid, model.predict_proba(X_valid)[:,
1]),
                roc_auc_score(y_test, model.predict_proba(X_test)[:,
1])],
    4))

display(expLog)

print("\n\n-----Historical Experiment Results......")
### Stroing the logs to file storage in case of kernel failure....
historical_logs = os.path.join(DATA_DIR, "expLog.csv")
if os.path.exists(historical_logs):
    old_explog = pd.read_csv(historical_logs)
    df = pd.concat([old_explog, expLog])
    df.drop_duplicates(inplace=True)
    df.to_csv(historical_logs, index=False)
else:
    expLog.to_csv(historical_logs, index=False)

display(pd.read_csv(historical_logs).sort_values(by="Test AUC"))
```

```
------------------------------------------------
X train           shape: (222176, 14)
X validation      shape: (46127, 14)
X test            shape: (39208, 14)
X X_kaggle_test   shape: (48744, 14)
Y train           shape: (222176,)
Y validation      shape: (46127,)
Y test            shape: (39208,)
-----------------------------------------

 Grid Search......


Fitting 5 folds for each of 3 candidates, totalling 15 fits
The best roc_auc_score is: 0.7350236784845827
------ The best parameters are: {'clf': LogisticRegression()}
The accuracy score of this model is:0.92
```

```
Grid search Results:-----

                          params  rank_test_score  mean_train_score  \
0  {'clf': LogisticRegression()}                1          0.735746
1        {'clf': BernoulliNB()}                2          0.685495
2      {'clf': SGDClassifier()}                3          0.625998

   std_train_score  mean_test_score  std_test_score
0         0.000825         0.735024        0.003488
1         0.002775         0.684591        0.002872
2         0.078512         0.622595        0.079653

------------------------------------------------------------
Experiment results so far......

                         exp_name  Train Acc  Valid Acc  Test  Acc
Train AUC  \
0  Gridserach_baseline14_features     0.9198     0.9193     0.9159
0.7357

   Valid AUC  Test AUC
0     0.7356    0.7356



-----Historical Experiment Results......

                         exp_name  Train Acc  Valid Acc  Test  Acc
Train AUC  \
0  Gridserach_baseline14_features     0.9198     0.9193     0.9159
0.7357

   Valid AUC  Test AUC
0     0.7356    0.7356

CPU times: user 10.7 s, sys: 3.08 s, total: 13.8 s
Wall time: 35.7 s
```

Evaluation metrics

Submissions are evaluated on area under the ROC curve between the predicted probability and the observed target.

The SkLearn `roc_auc_score` function computes the area under the receiver operating characteristic (ROC) curve, which is also denoted by AUC or AUROC. By computing the area under the roc curve, the curve information is summarized in one number.

```
from sklearn.metrics import roc_auc_score
>>> y_true = np.array([0, 0, 1, 1])
>>> y_scores = np.array([0.1, 0.4, 0.35, 0.8])
```

```
>>> roc_auc_score(y_true, y_scores)
0.75
```

AUC score

```
from sklearn.metrics import roc_auc_score, roc_curve
roc_auc_score(y_train, model.predict_proba(X_train)[:, 1])
```

Submission File Prep

For each SK_ID_CURR in the test set, you must predict a probability for the TARGET variable. The file should contain a header and have the following format:

```
SK_ID_CURR,TARGET
100001,0.1
100005,0.9
100013,0.2
etc.
```

```
test_class_scores = model.predict_proba(X_kaggle_test)[:, 1]
```

```
test_class_scores[0:10]
```

```
# Submission dataframe
submit_df = datasets["application_test"][['SK_ID_CURR']]
submit_df['TARGET'] = test_class_scores

submit_df.head()
```

```
submit_df.to_csv("submission.csv",index=False)
```

Kaggle submission via the command line API

```
! kaggle competitions submit -c home-credit-default-risk -f
submission.csv -m "baseline submission"
```

report submission

Click on this link

image.png

Write-up

For this phase of the project, you will need to submit a write-up summarizing the work you did. The write-up form is available on Canvas (Modules-> Module 12.1 - Course Project - Home Credit Default Risk (HCDR)-> FP Phase 2 (HCDR) : write-up form). It has the following sections:

Abstract

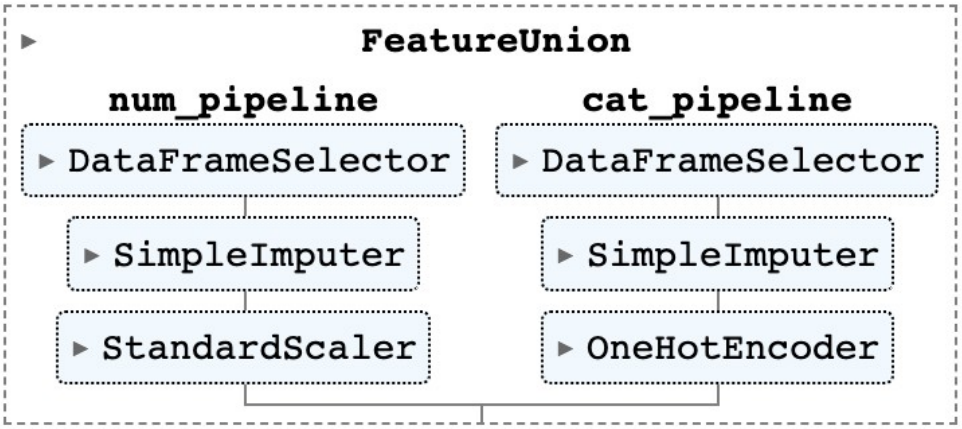
Data analysis is one of the most important and essential aspects of Machine Learning. Building Machine Learning Models is ineffective if you don't understand the data. We know that Feature Engineering is at the foundation of every Machine Learning model, and that if we can't make sense of the data, we won't be able to construct the explanatory features that our models would use for classification. Exploratory Data Analysis is the process of looking at data to get to the core of it and finding patterns, behaviors, dependencies, and anomalies, as well as testing hypotheses and generating summaries using statistical and graphical tools. In this Phase 1 of the project, we have downloaded the data from Kaggle. Few EDA techniques are done on the given datasets. This has been helpful to find out the missing values for each attribute and also in deciding important attributes. By leveraging insights from EDA we have done basic feature engineering.

Introduction

Feature Engineering and transformers

We one-hot encoded all the category features for Feature Engineering. Pipelines for all secondary tables are created. The aggregated data is merged in main table using pipeline.Numerical pipeline and categorical pipeline are specified as shown.

##Feature Engineering :



##Pipelines:

data_prep_pipeline = FeatureUnion(transformer_list=[("num_pipeline", num_pipeline), ("cat_pipeline", cat_pipeline),])

The above pipeline is used for aggregating the categorical and numerical data

And the code below will use some of the features after being preprocessed through the pipeline. The baseline model is a logistic regression model

full_pipeline_with_predictor = Pipeline([("preparation", data_prep_pipeline), ("linear", LogisticRegression())])

Experimental results

```
 Grid Search......


Fitting 5 folds for each of 3 candidates, totalling 15 fits
The best roc_auc_score is: 0.7350236784845827
------ The best parameters are: {'clf': LogisticRegression()}
The accuracy score of this model is:0.92


Grid search Results:-----
```

	params	rank_test_score	mean_train_score	std_train_score	mean_test_score	std_test_score
0	{'clf': LogisticRegression()}	1	0.735746	0.000825	0.735024	0.003488
1	{'clf': BernoulliNB()}	2	0.685495	0.002775	0.684591	0.002872
2	{'clf': SGDClassifier()}	3	0.625998	0.078512	0.622595	0.079653

```
------------------------------------------------------------
Experiment results so far......
```

	exp_name	Train Acc	Valid Acc	Test Acc	Train AUC	Valid AUC	Test AUC
0	Gridserach_baseline14_features	0.9198	0.9193	0.9159	0.7357	0.7356	0.7356

```
-----Historical Experiment Results......
```

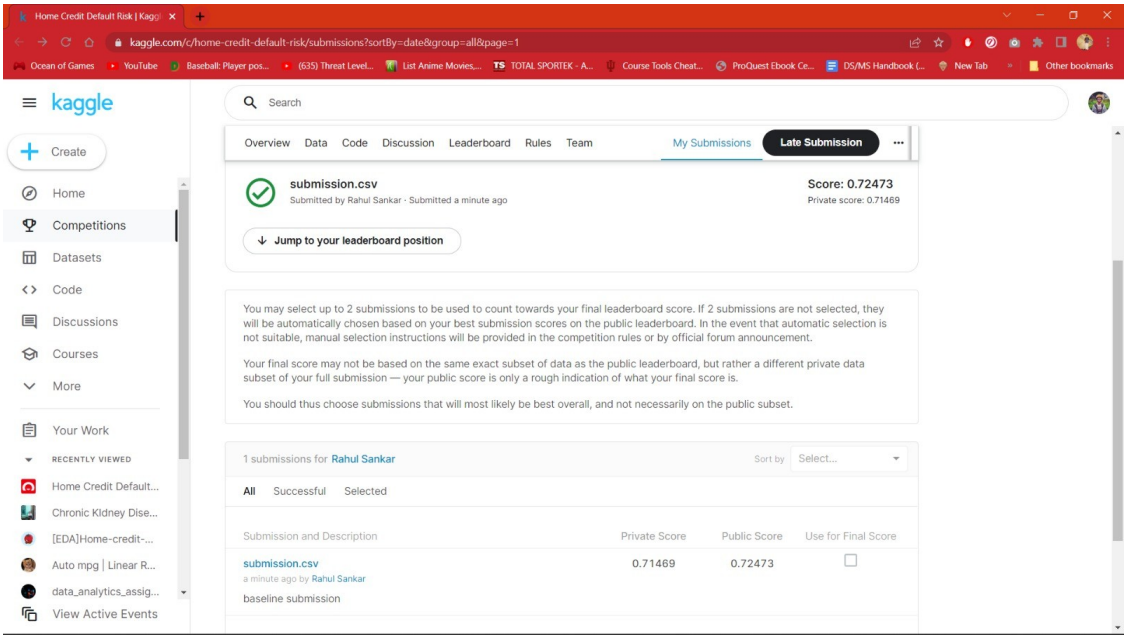
	exp_name	Train Acc	Valid Acc	Test Acc	Train AUC	Valid AUC	Test AUC
0	Gridserach_baseline14_features	0.9198	0.9193	0.9159	0.7357	0.7356	0.7356

```
CPU times: user 10.7 s, sys: 3.08 s, total: 13.8 s
Wall time: 35.7 s
```

Discussion

In this phase we have tried and gained some deeper knowledge regarding the datasets and the relationships between the datasets. The process of EDA has been helpful to know more about the relationships between various features. We built a baseline model where some of the features are used after being preprocessed through the pipeline. Plans for Phase 2: In phase two we are going to focus a bit more on the feature engineering Like engineering RFM features(Recency, Frequency, and Monetary value) And also building a multi-layer perception (MLP) model in PyTorch.

Kaggle Submission



References

Some of the material in this notebook has been adopted from here

TODO: Predicting Loan Repayment with Automated Feature Engineering in Featuretools

Read the following:

- feature engineering via Featuretools library:
  - https://github.com/Featuretools/predict-loan-repayment/blob/master/Automated%20Loan%20Repayment.ipynb
- https://www.analyticsvidhya.com/blog/2018/08/guide-automated-feature-engineering-featuretools-python/
- feature engineering paper: https://dai.lids.mit.edu/wp-content/uploads/2017/10/DSAA_DSM_2015.pdf
- https://www.analyticsvidhya.com/blog/2017/08/catboost-automated-categorical-data/