*Group 2: Fabiana Caccavale, Matteo Gioia, Martina Manno, Vincenzo J. Striano, Marco Vita Antonio, Marco Pisano **

Ethics for AI project: Potential biases in AI-based financial services

Libraries with fixed versions and data loading

In [ ]:
```
!conda install --yes python==3.7.10
!conda install --yes xgboost==1.6.2
!conda install --yes shap==0.41.0
!conda install --yes pandas==1.3.5
!conda install --yes plotly==5.10.0
!conda install --yes scikit-learn==1.0.2
!conda install --yes matplotlib
!conda install --yes seaborn
!conda install --yes dython
```

In [1]:
```
import xgboost as xgb
import pandas as pd
import numpy as np
import shap
import plotly.express as px
import matplotlib.pyplot as plt
import seaborn as sns
import dython
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.preprocessing import OneHotEncoder
from dython.nominal import associations
from plotly.offline import iplot, init_notebook_mode
init_notebook_mode(connected = True)
from IPython.display import Image
```

The project (https://www.kaggle.com/c/home-credit-default-risk) is inspired by a kaggle competition focused on predicting the future payment behaviour of clients from a loan application.

The dataset considered is the "application train" and from this, 26 columns (1 target, 1 ID and 24 features) were selected out of a total of 122 in order to have a more interpretable overview.

In [2]:
```
ap_train = pd.read_csv('application_train.csv',
                       usecols =  ['TARGET', 'SK_ID_CURR', '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', 'OCCUPATION_TYPE',
                                   'NAME_FAMILY_STATUS', 'NAME_HOUSING_TYPE', 'REGION_POPULATION_RELATIVE', 'DAYS_BIRTH',
                                   'DAYS_EMPLOYED', 'DAYS_REGISTRATION', 'DAYS_ID_PUBLISH', 'OWN_CAR_AGE', 'EXT_SOURCE_1',
                                   'EXT_SOURCE_2', 'EXT_SOURCE_3'])
print(f"Application train dataset shape: {ap_train.shape}")
```

```
Application train dataset shape: (307511, 26)
```

In [3]:
```
# DAYS_BIRTH is transformed in AGE
ap_train['AGE'] = -round(ap_train['DAYS_BIRTH']/365)
ap_train.drop(columns='DAYS_BIRTH', inplace=True)
```

[#1] Interesting variables

The first step is to reflect on which variables might be prone to an ethical discussion.

In anti-discrimination philosophy, Cass Sunstein writes, *"without good reason, social and legal structures should not turn differences that are both highly visible and irrelevant from the moral point of view into systematic social disadvantages"* (Sunstein 1994).

Algorithmic-related bias refers to systematic and repeatable errors in a mathematical or computer system that leads to 'unfair' outputs, privileging one or more groups over others.

The variables that can lead to unjust judgment due to: historical bias representation bias measurement bias have been labelled as ethical.

1. **CODE_GENDER**: results in ethical relevance because an individual's gender should not be a useful variable in the choice of granting a mortgage (Men vs. Women, Men vs. Transgender)

2. **FLAG_OWN_CAR**: results in ethical relevance because the ownership of a car does not affect the choice of granting a mortgage. Owning or not owning a car does not turn out to be related to a customer's creditworthiness. It is more than plausible, that individuals living in a large city prefer to use public or sustainable transportation.

3. **FLAG_OWN_REALTY**: homeownership does not affect the choice of granting a mortgage. Owning or not owning a car does not appear to be related to a client's creditworthiness. Although home ownership is commonly associated with large economic capacity, statistics show that in countries with higher GDP, the percentage of individuals owning a home for residential use decreases. For example, Romania does not rank as a high-income economy within the European Union, has the highest homeownership rates in the EU, and has the most crowded housing. Romania has 96.4 percent of its population owning homes.

4. **CNT_CHILDREN**: this turns out to be of ethical relevance because the number of children does not affect the choice to grant a mortgage. There is a tendency to think that a large number of children results in large expenses that decrease the ability to repay the debt. First, relying only on the number of offspring does not indicate how many more of the children are dependent on the individual; they may already be financially independent. Moreover, it is incorrect not to take into consideration that a possible partner might contribute to the expenses of supporting the offspring. Therefore, we believe that other variables should also be taken into consideration.

5. **NAME_TYPE_SUITE**: who accompanies the borrower should not be considered in the granting of a mortgage. The one who should be solvent is the one who makes the commitment. The judgment cannot be influenced by the accompanying person under any circumstances. The judgment must be based on the borrower.

6. **NAME_INCOME_TYPE**: This variable is prone to ethical discourse in part. It is not fair to make a judgment about solvency between an individual state servant and a commercial associate or pensioner for example. In both cases, the value should not bias the judgment and thus is of ethical relevance. However, in this variable, we also find the unemployment factor, which in our view cannot be considered ethically relevant. The fact that an individual is unemployed is a strong factor that should condition the individual's judgment of insolvency. The fact that an individual does not have a fixed salary and therefore economic insecurity in theory should be a very relevant factor for the bank.

7. **NAME_EDUCATION_TYPE**: turns out to be ethically relevant because the level of education should not be taken into consideration in the granting of a mortgage (Although a high level of education may indicate possible higher earnings, it still does not ensure economic stability or a lower level of education does not undermine an individual's creditworthiness).

8. **OCCUPATION_TYPE**: The type of occupation is not an accurate index of income. Even in the same category of occupation, there can be a huge income gap. A chef in a starred restaurant will not have the same income as a chef in a chain restaurant.

9. **NAME_FAMILY_STATUS**: this turns out to be ethically relevant because family status should not be taken into consideration when granting a mortgage. The number of individuals in the family does not ensure greater economic availability. Two married individuals might still have less aggregate income than a single individual or a widow.

10. **NAME_HOUSING_TYPE**: The type of property should not be taken into consideration when granting a mortgage. Homeownership does not indicate greater economic availability and vice versa. Young people might be discriminated against because, especially in Italy, it is not common to own a home. Those who travel a lot and therefore have economic availability may not have home ownership.

11. **REGION_POPULATION_RELATIVE**: We consider it of ethical relevance as well as absolute irrelevance toward judgment since a highly populated region is not an indication of wealth or poverty (Rio de Janeiro vs. New York City). Judging also on the basis of the regional population may be unnecessary as well as incorrect.

12. **DAYS_BIRTH**: this variable discriminates against young people because they are considered less creditworthy or the elderly because they are subject to lower life expectancy.

13. **DAYS_EMPLOYED**: it is not necessarily the case that someone who is a new employee has a lower salary, and therefore potentially less creditworthy, or that someone who has been employed for a long time has more disposable income.

14. **DAYS_ID_PUBLISH**: this variable is more important for a discourse of verifying the identity of the individual and thus in the validity of the loan issuance rather than the solvency of the individual.

15. **OWN_CAR_AGE**: The age of a car cannot be an indication of the individual's solvency. A dated car might be worth more than a new car, and an individual might keep a car as a matter of affection, and this is not an indication of the individual's poverty or wealth.

[#2] Interesting variables with respect to target

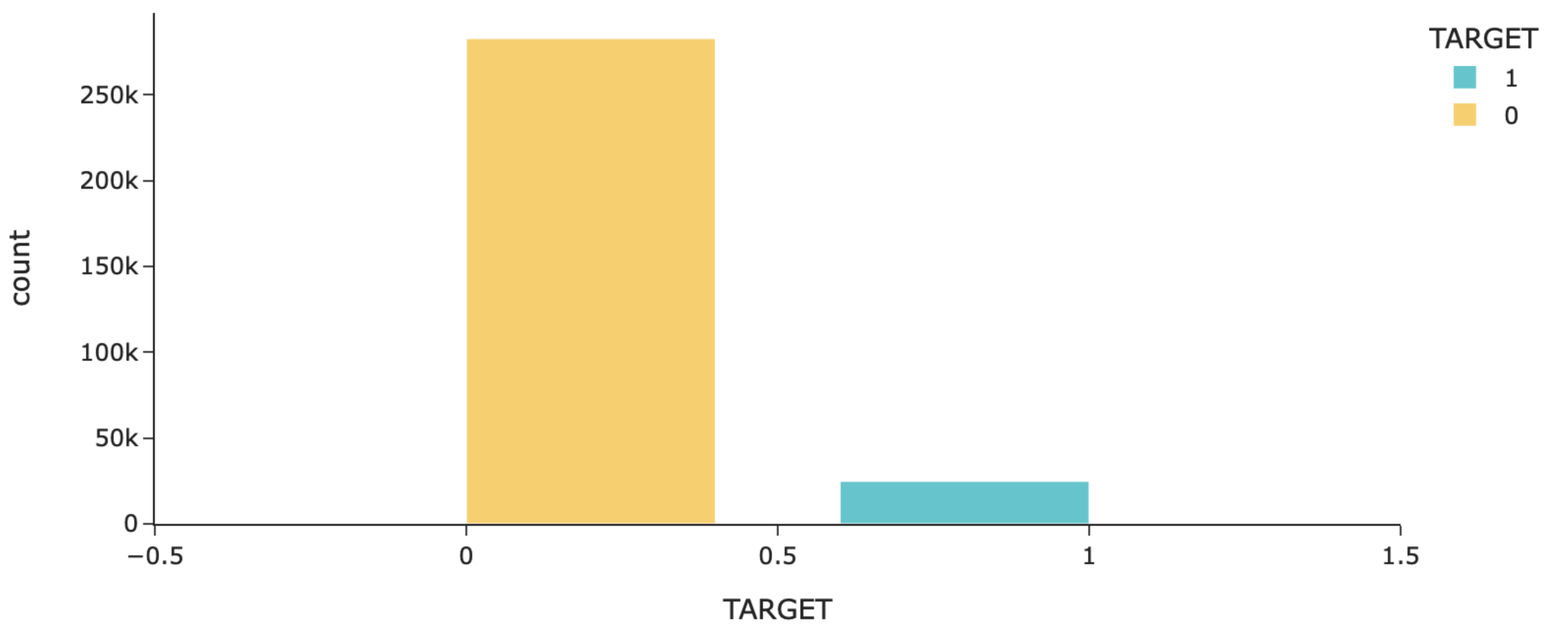
The second step concerns the understanding of the ethical variables highlited with respect to the target variable.

TARGET

In [4]:
```
ap_train.groupby('TARGET').apply(len)
```

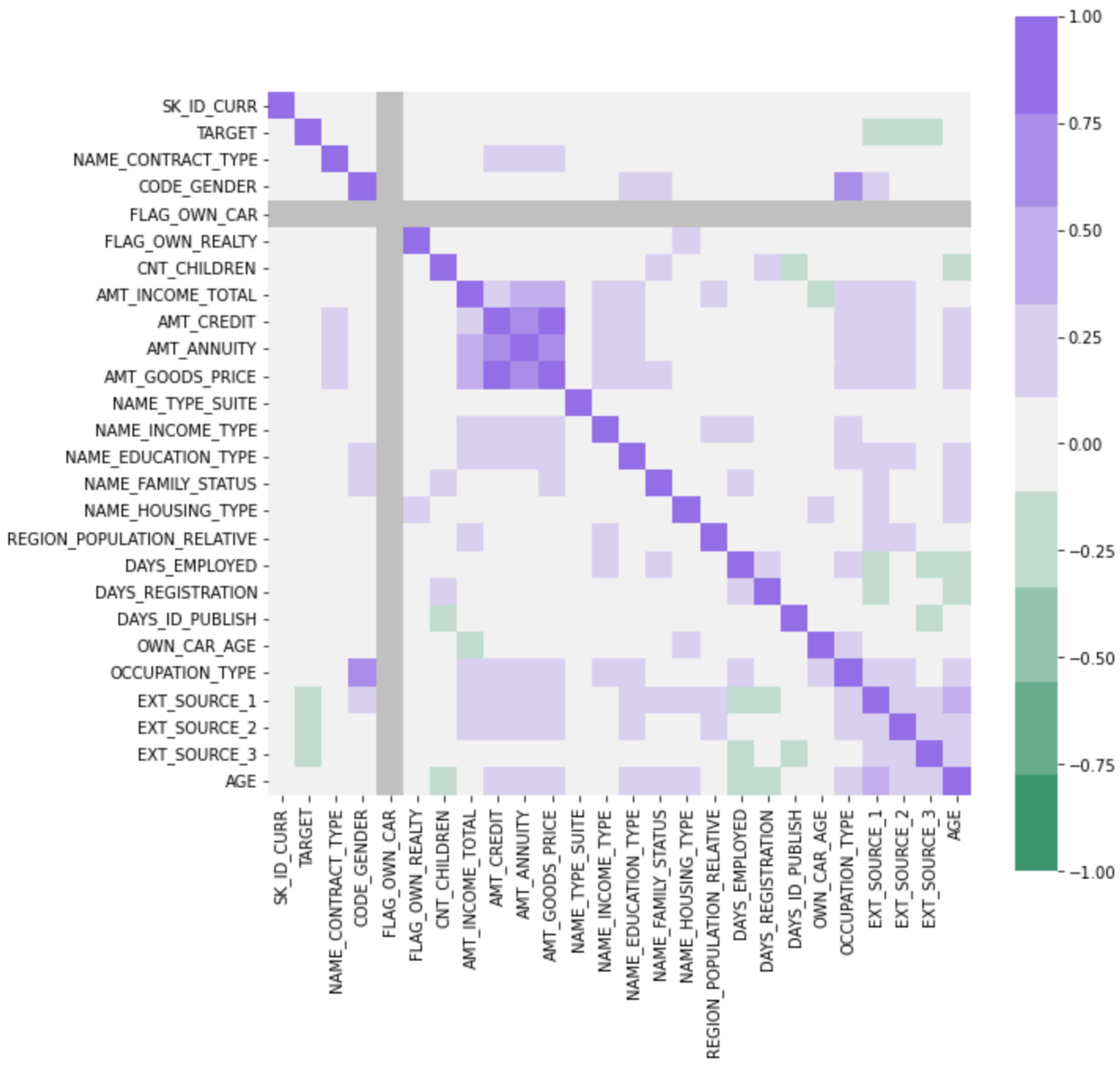
Out[4]:
```
TARGET
0    282686
1     24825
dtype: int64
```

In [5]:
```
fig = px.histogram(ap_train, width=800, height=400, color="TARGET", x="TARGET", barmode="group",template="simple_white", color_discrete_sequence
fig.update_layout(xaxis=dict(showgrid=False, ), yaxis=dict(showgrid=False))
fig.show()
```



The target is a binary variable with no missing values. It can assume the value 1 whether the client has payment difficulties (such as late payment of more than X days on at least one of the first Y instalments of the loan); or 0 in all other cases. The overall number of clients is about 307.511 and the graph above shows that only 24.825 clients (8% of the total) have been categorized among those who have payment difficulties.

In [6]:
```
complete_correlation= associations(ap_train, figsize=(10,10), nan_strategy = 'drop_samples', vmin=-1, vmax=1,
                                   cmap=sns.diverging_palette(150, 275, s=80, l=55, n=9),  annot= False)
```

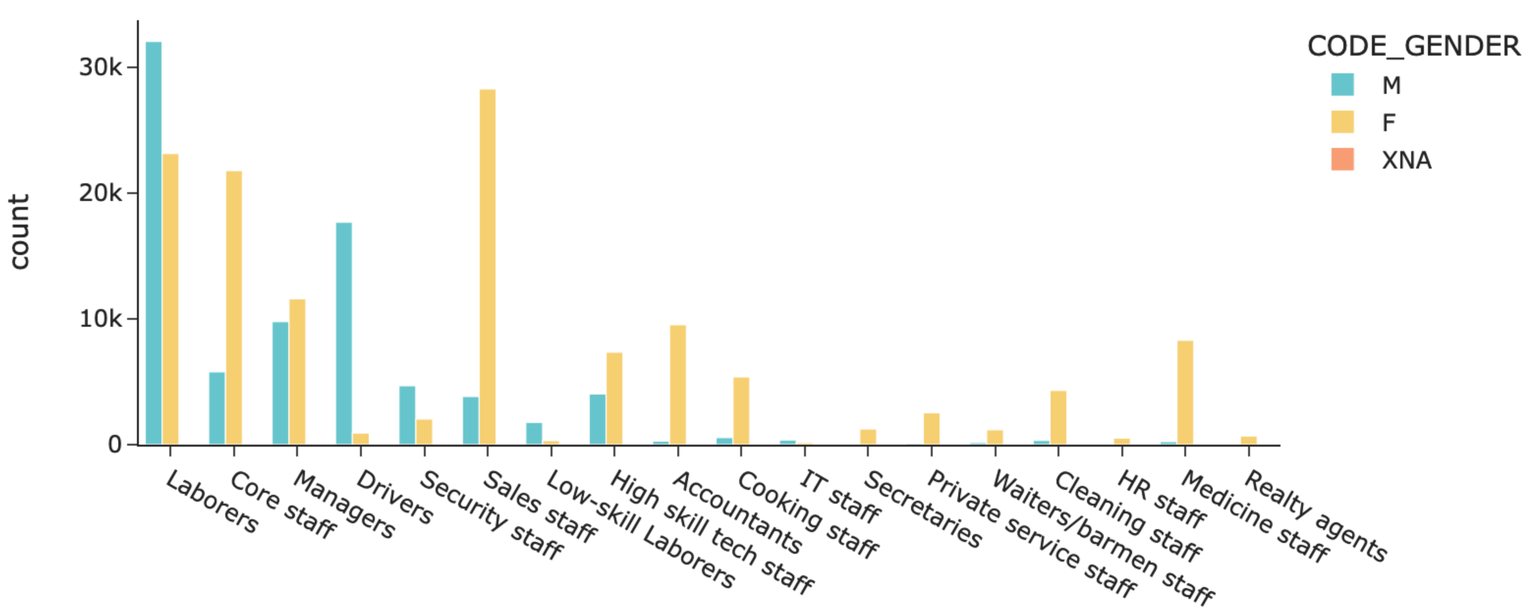


The target variable appears to be not correlated with any of the other dataset's variables. There is only a slight negative correlation with the variables EXT_SOURCE_1, EXT_SOURCE_2, and EXT_SOURCE_3 (at which the correlation is respectively -0.14, -0.14, and -0.17). A possible interpretation is that clients who have a low external source score would be inclined to be in the target number 1, then in the group of clients with some payment difficulties. However, since it is not known whether the normalized score from the external data source, to which these variables refer, has an ethical nature or not, it is not possible to determine whether these variables have ethical relevance.

Despite the fact that the financial institution has requested a lot of data, many of which are extremely sensitive, the architecture of the model allows it to determine for itself which of the variables are relevant or not. According to the model, the relevant ones are unethical and mostly come from third-party sources, over which we have no knowledge and the bank no control.

Another consideration can be done in terms of one of the variables that we have considered most prone to the ethical discussion: this is the code gender. This variable seems to be correlated with another ethical variable which is occupation type.

In [7]:
```
fig = px.histogram(ap_train,width=800, height=400, color="CODE_GENDER", x="OCCUPATION_TYPE", barmode="group",template="simple_white", color_disc
fig.update_layout(xaxis=dict(showgrid=False, ), yaxis=dict(showgrid=False))
fig.show()
```



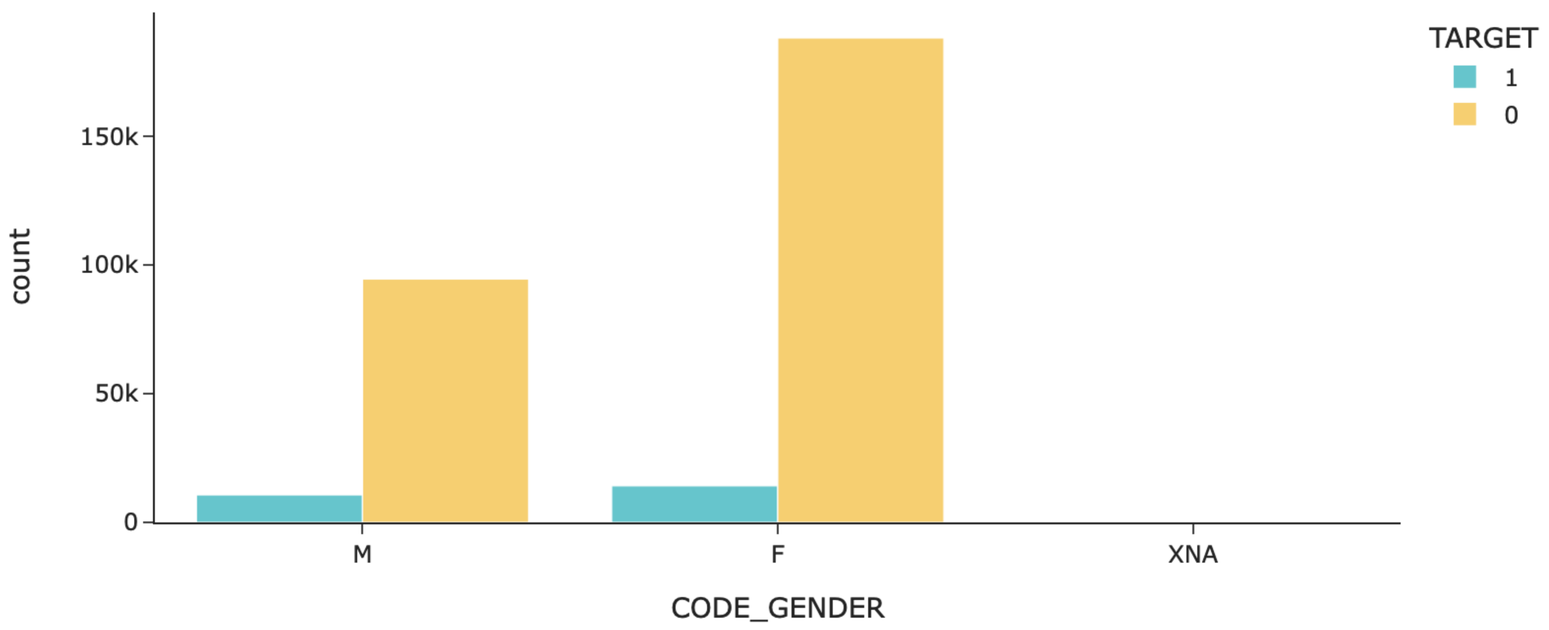
As shown in the graph, there are some occupation types, such as drivers, in which 95% is male, and on the other hand some occupations, such as the medicine staff, in which 97% is female.

CODE_GENDER

In [8]:
```
ap_train.groupby('CODE_GENDER').apply(len)
```

Out[8]:
```
CODE_GENDER
F      202448
M      105059
XNA         4
dtype: int64
```

In [9]:
```
fig = px.histogram(ap_train,width=800, height=400, color="TARGET", x="CODE_GENDER", barmode="group",template="simple_white", color_discrete_sequ
fig.update_layout(xaxis=dict(showgrid=False, ), yaxis=dict(showgrid=False))
fig.update_layout(paper_bgcolor='rgba(0,0,0,0)',plot_bgcolor='rgba(0,0,0,0)')
fig.show()
```



The variable indicates the gender of the client applying for the loan. The sample is composed of 65% women. Out of the total number of clients, there are 4 who preferred not to declare their gender, this could be due to a random reason or because it might be perceived by the client as sensible data. Moreover, this variable appears to be not relevant in the payment behaviours of clients because it cannot in any way be generalized that women or men are more or less solvent. The variable allows for several reflections:

- What would this same graph have looked like 50 years ago?
- Would the percentage of women in the total have been as high?
- Would the percentage of women with payment difficulties have been higher?
- If so, could these steps forward be due to measures to reduce the gender pay gap?

FLAG_OWN_CAR

In [10]:
```
ap_train.groupby('FLAG_OWN_CAR').apply(len)
```

Out[10]:
```
FLAG_OWN_CAR
N    202924
Y    104587
dtype: int64
```

In [11]:
```
fig = px.histogram(ap_train, width=800, height=400,color="TARGET", x="FLAG_OWN_CAR", barmode="group", template="simple_white", color_discrete_se
fig.update_layout(xaxis=dict(showgrid=False, ), yaxis=dict(showgrid=False))
fig.show()
```

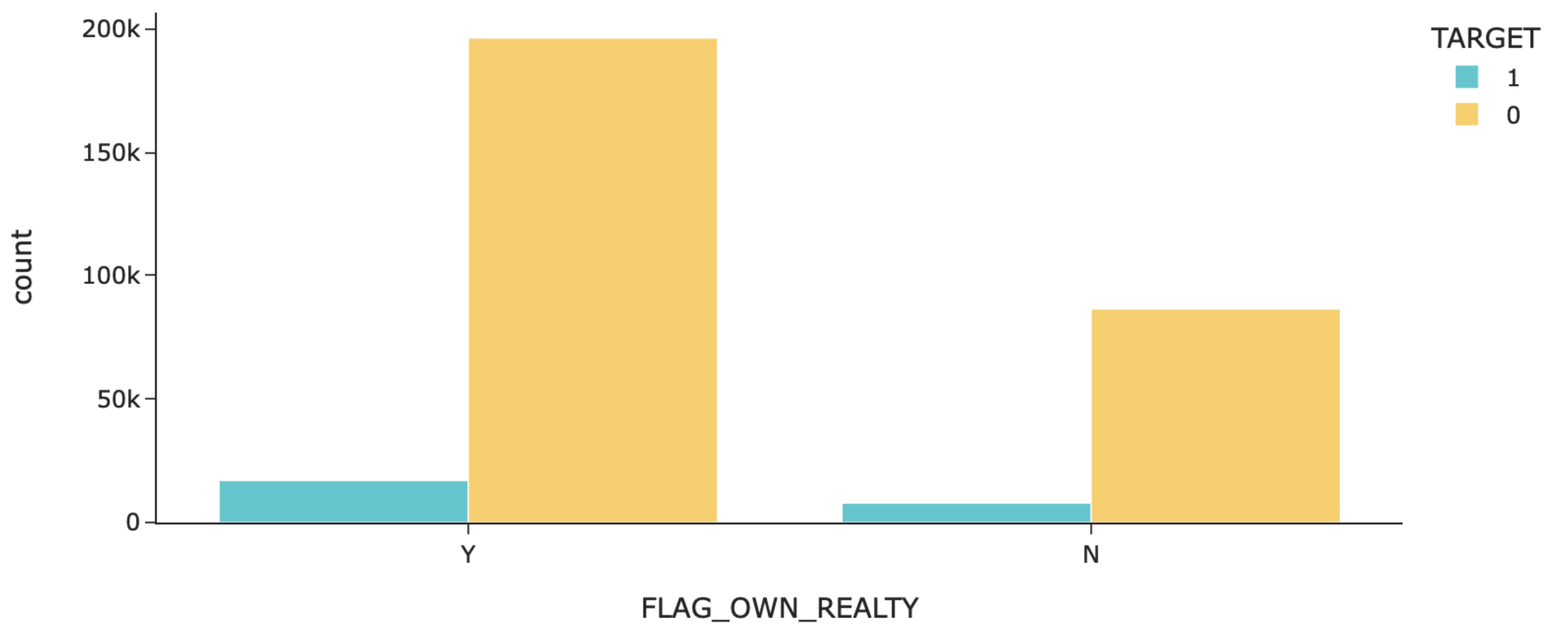
The binary variable indicates whether the client owns a car or not. Out of the total number of clients, 65% declare to not own a car. This variable is not relevant with regard to the solvency of clients because, as shown in the graph, among those who do not own a car, 8% have payment difficulties but also among those who own a car, the 7% have solvency problems.

FLAG_OWN_REALTY

In [12]:
```
ap_train.groupby('FLAG_OWN_REALTY').apply(len)
```

Out[12]:
```
FLAG_OWN_REALTY
N     94199
Y    213312
dtype: int64
```

In [13]:
```
fig = px.histogram(ap_train, width=800, height=400,color="TARGET", x="FLAG_OWN_REALTY", barmode="group", template="simple_white", color_discrete
fig.update_layout(xaxis=dict(showgrid=False, ), yaxis=dict(showgrid=False))
fig.show()
```



The binary variable indicates whether the client owns a house or a flat or not. Out of the total number of clients, 70% declare to own a house/flat. The graph above shows that this variable is quite irrelevant concerning the solvency of clients. In fact, among those who own realty, 8% have payment difficulties but also among those who do not own realty, the 8% have payment difficulties.

CNT_CHILDREN

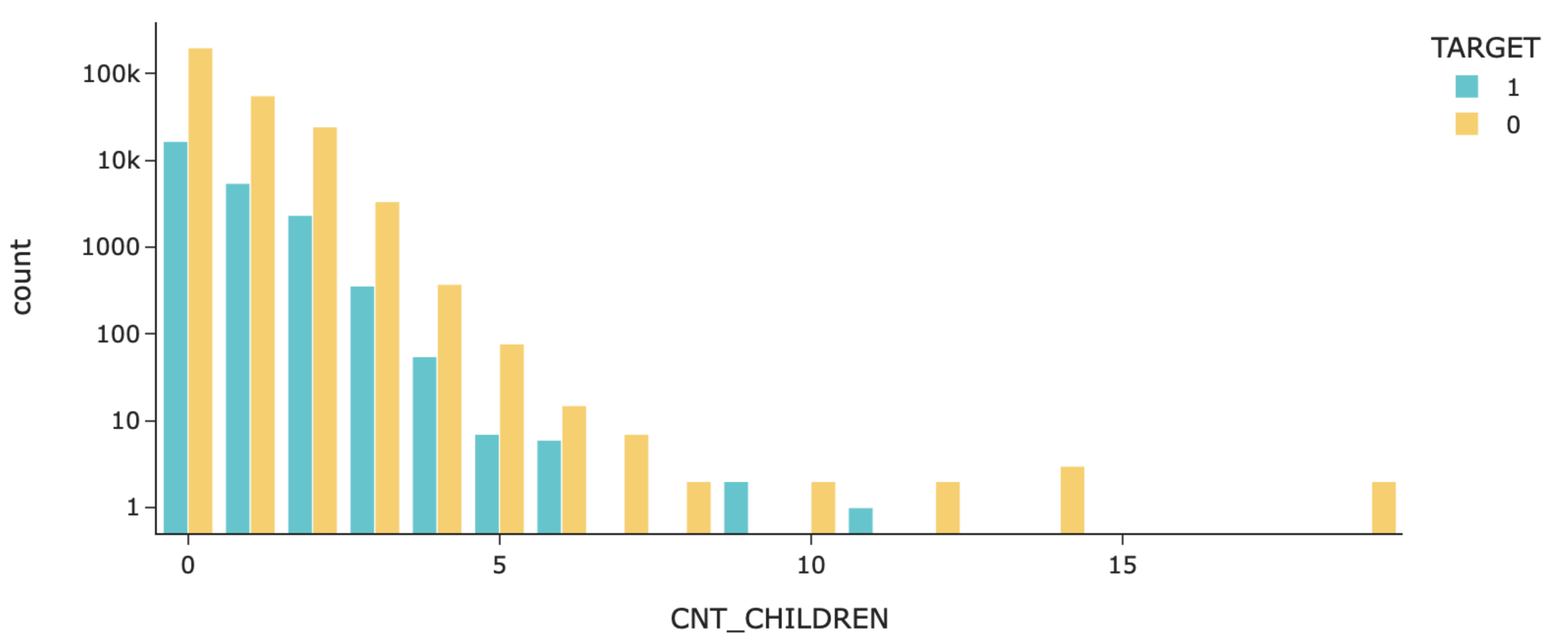
In [14]:
```
ap_train.groupby('CNT_CHILDREN').apply(len)
```

Out[14]:
```
CNT_CHILDREN
0     215371
1      61119
2      26749
3       3717
4        429
5         84
6         21
7          7
8          2
9          2
10         2
11         1
12         2
14         3
19         2
dtype: int64
```

In [15]:
```
ap_train['CNT_CHILDREN'].describe()
```

Out[15]:
```
count    307511.000000
mean          0.417052
std           0.722121
min           0.000000
25%           0.000000
50%           0.000000
75%           1.000000
max          19.000000
Name: CNT_CHILDREN, dtype: float64
```

In [16]:
```
fig = px.histogram(ap_train, width=800, height=400,color="TARGET", x="CNT_CHILDREN", barmode="group", log_y = True, template="simple_white", col
fig.update_layout(xaxis=dict(showgrid=False, ), yaxis=dict(showgrid=False))
fig.show()
```



The variable represents the number of children the client applying for the loan has. The number of children goes from 0 to 19. There are 215.371 over 307.511 clients who declare that they do not have children. Among those who do not have children, 92% are solvent clients. Among clients who have from 1 to 6 children, at least, 70% are solvent clients. What is evident is that there are some clients who have 7,8,10,12,14,19 children who are all among solvent clients but take into consideration that they are only 18 over the total amount of clients. On the other hand, clients who have 9 and 11 children are all with payment difficulties but they are overall only 3 clients. It is possible to conclude that clients who have 0 or at least 1 child represent 75% of the overall amount of clients and respectively 92% and 91% of them are solvent.

NAME_TYPE_SUITE

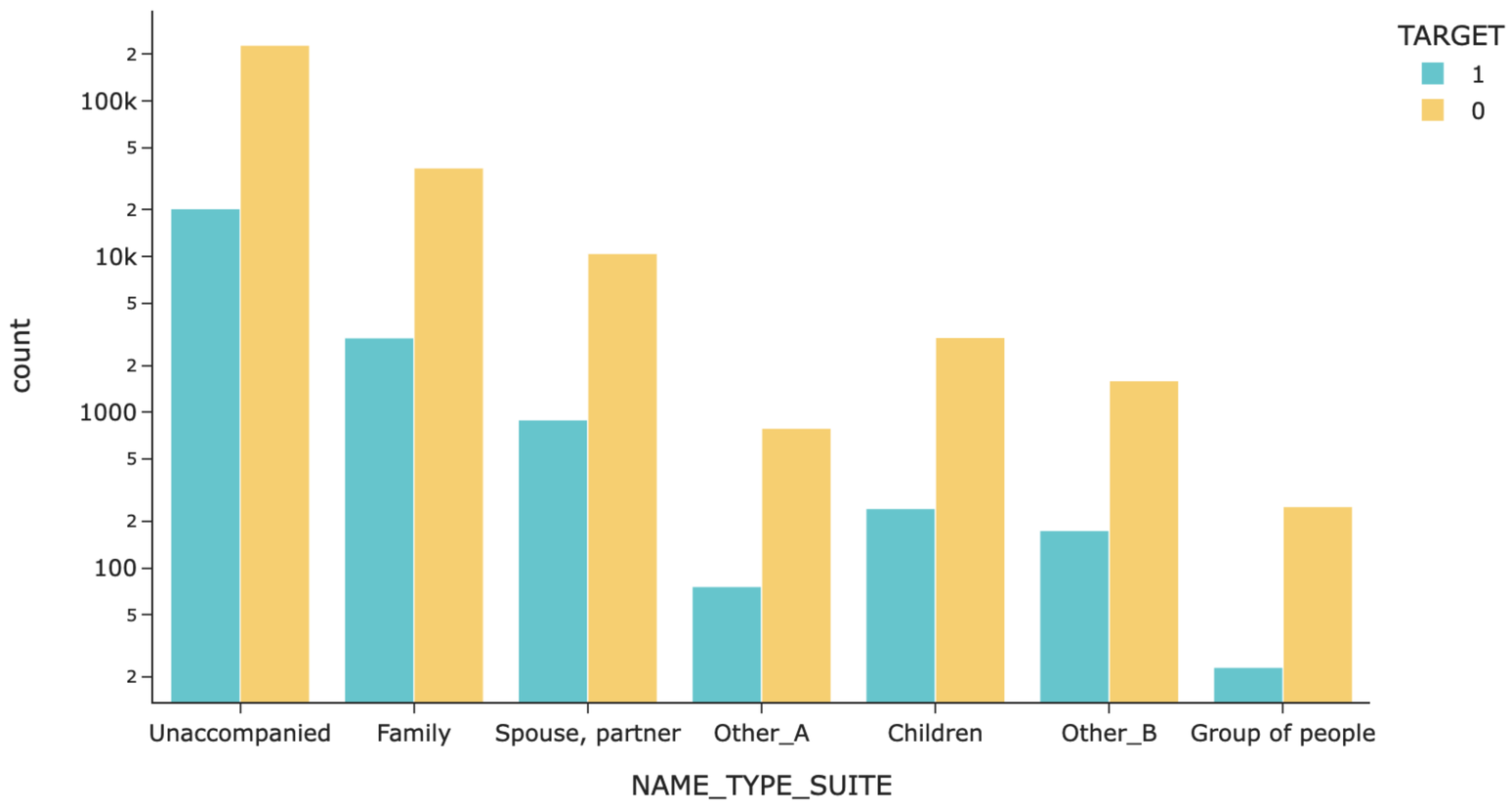
In [17]:
```
ap_train['NAME_TYPE_SUITE'].isnull().sum()
```

Out[17]:
```
1292
```

In [18]:
```
ap_train.groupby('NAME_TYPE_SUITE').apply(len)
```

Out[18]:
```
NAME_TYPE_SUITE
Children             3267
Family              40149
Group of people       271
Other_A               866
Other_B              1770
Spouse, partner     11370
Unaccompanied      248526
dtype: int64
```

In [19]:
```
fig = px.histogram(ap_train,width=800, height=500, color="TARGET", x="NAME_TYPE_SUITE", barmode="group",log_y=True, template="simple_white", col
fig.update_layout(xaxis=dict(showgrid=False, ), yaxis=dict(showgrid=False))
fig.show()
```



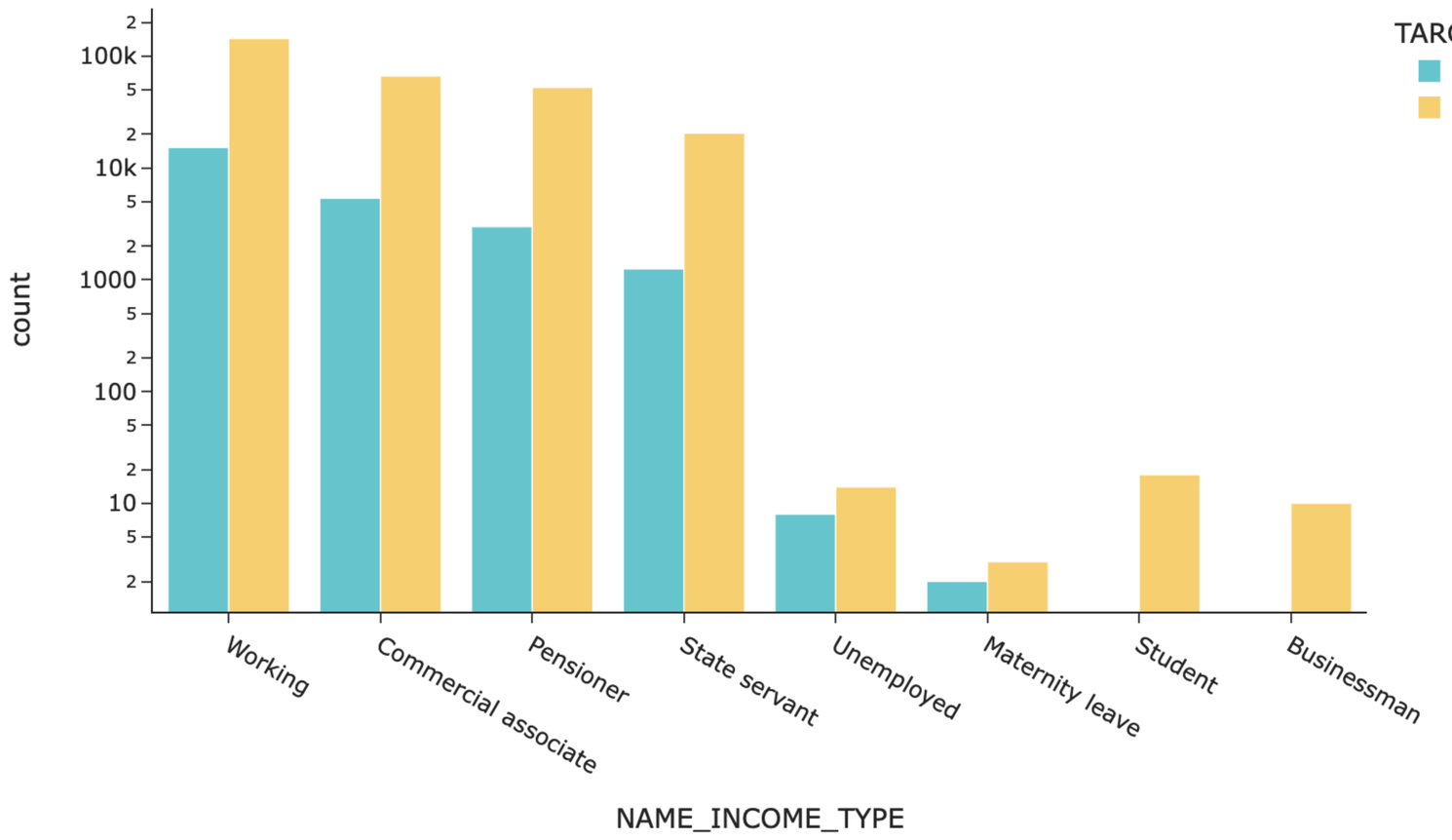
The variable indicates who was accompanying the client when applying for the loan. There are 1292 missing values which refers to clients who choose not to declare who they are accompanied by. The most frequent type of this variable is "unaccompanied" which covers more or less 80% of cases. As shown by the graph above, there is not a category that has an influence on the target variable, in fact, all of them have at least 90% of solvent clients.

NAME_INCOME_TYPE

In [20]:
```
ap_train.groupby('NAME_INCOME_TYPE').apply(len)
```

Out[20]:
```
NAME_INCOME_TYPE
Businessman                10
Commercial associate     71617
Maternity leave              5
Pensioner                55362
State servant            21703
Student                     18
Unemployed                  22
Working                 158774
dtype: int64
```

In [21]:
```
fig = px.histogram(ap_train, width=800, height=500,color="TARGET", x="NAME_INCOME_TYPE", barmode="group",template="simple_white", log_y=True, co
fig.update_layout(xaxis=dict(showgrid=False, ), yaxis=dict(showgrid=False))
fig.show()
```



The variable indicates the source of the client's income:

- Working (51% of clients)
- Commercial associate (23% of clients)
- Pensioner (18% of clients)
- State servant (7% of clients)

- Student
- Maternity leave
- Businessman
- Unemployed

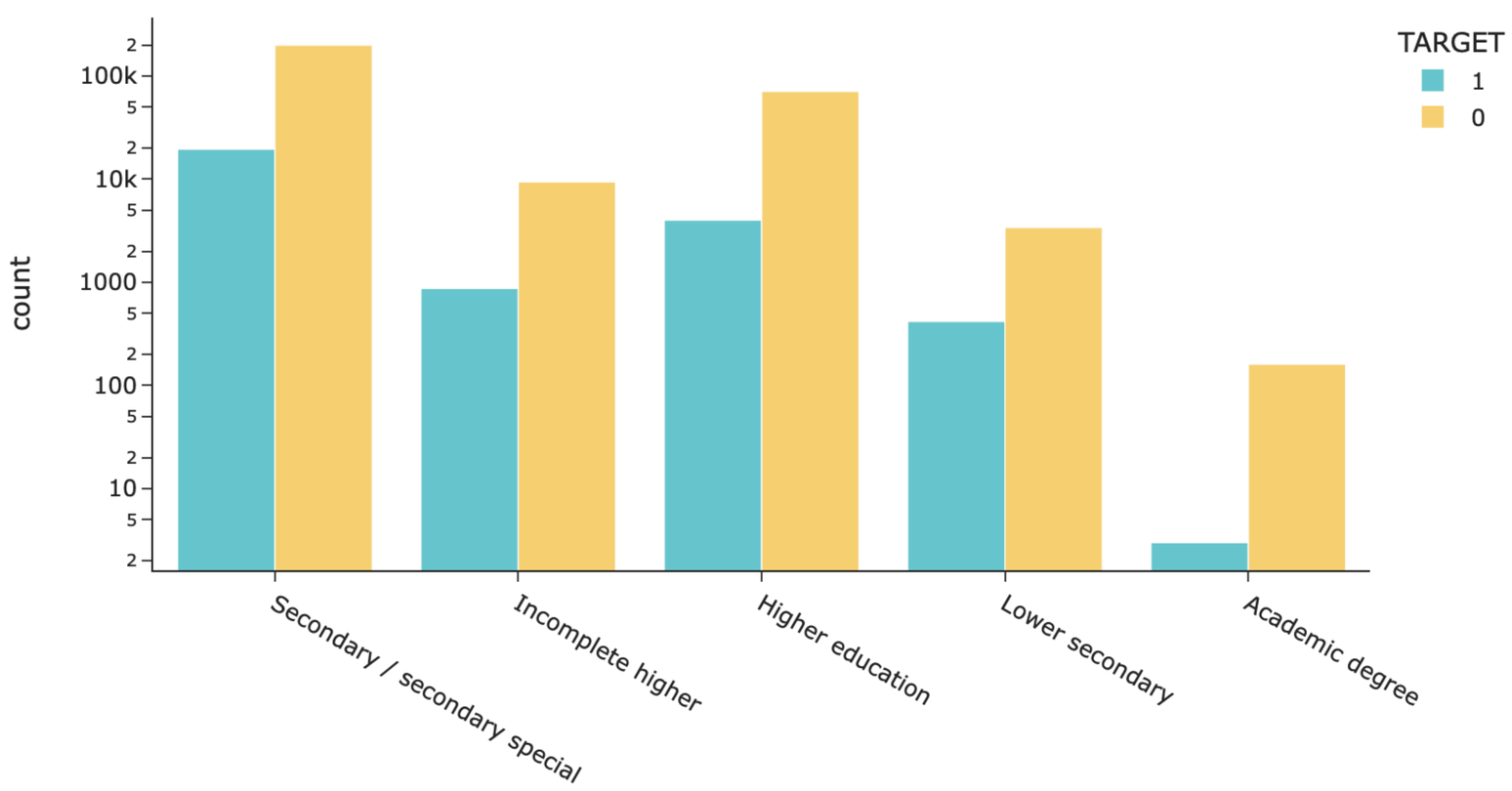
The most frequent source of the client's income is general work, the other main categories of income are made up of commercial associates, state servants, and pensioners. The last 1% of income type derives from students, businessmen, people who perceived maternity leave, and people unemployed. In this case, what is expected is fulfilled because students, unemployed persons, businessmen, and women on maternity leave are unlikely to apply for a loan for diametrically opposite reasons. For example, a student or an unemployed person will not apply for a loan because they are unlikely to be granted one and on the other hand, a businessman should not need a loan. One interesting thing that emerges from the graph is that both students and businessmen have no problems paying their loans. Whereas, 30% of unemployed (8 clients) have payment difficulties.

NAME_EDUCATION_TYPE

In [22]:
```
ap_train.groupby('NAME_EDUCATION_TYPE').apply(len)
```

Out[22]:
```
NAME_EDUCATION_TYPE
Academic degree                  164
Higher education               74863
Incomplete higher              10277
Lower secondary                 3816
Secondary / secondary special  218391
dtype: int64
```

In [23]:
```
fig = px.histogram(ap_train,width=800, height=500, color="TARGET", x="NAME_EDUCATION_TYPE", barmode="group", log_y=True,template="simple_white",
fig.update_layout(xaxis=dict(showgrid=False, ), yaxis=dict(showgrid=False))
fig.show()
```



The variable indicates the level of highest education the client achieved which can be categorized in:

- Secondary/Secondary special
- Higher education
- Incomplete higher
- Lower Secondary
- Academic degree.

71% of clients have a secondary level of education and 91% of them are solvent. All these categories report, at least, 89% of solvent clients. Therefore, this variable is not relevant to highlighted client categories in terms of solvency.

OCCUPATION_TYPE

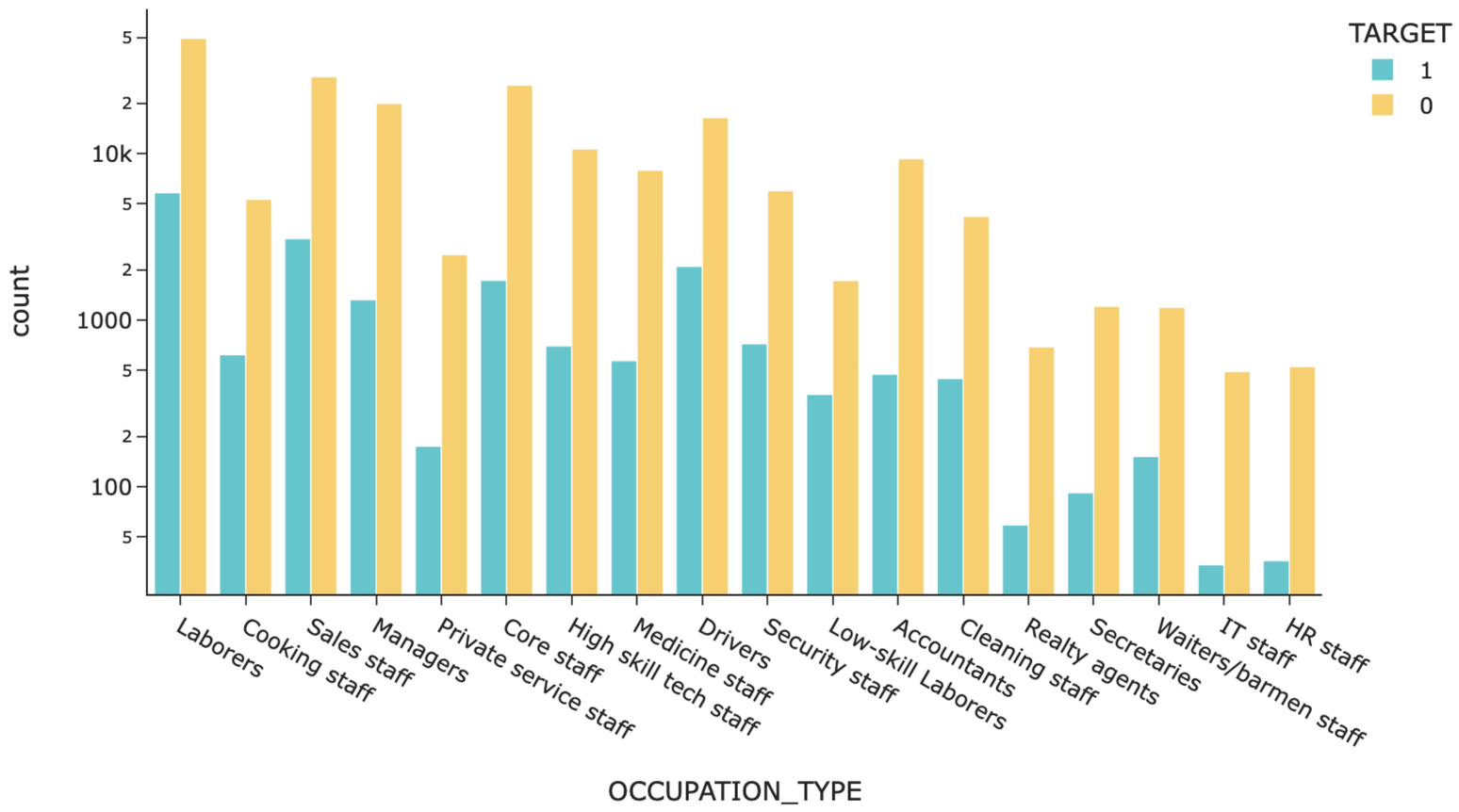
In [24]:
```
ap_train['OCCUPATION_TYPE'].isnull().sum()
```

Out[24]:
```
96391
```

In [25]:
```
ap_train.groupby('OCCUPATION_TYPE').apply(len)
```

Out[25]:
```
OCCUPATION_TYPE
Accountants             9813
Cleaning staff          4653
Cooking staff           5946
Core staff             27570
Drivers                18603
HR staff                 563
High skill tech staff  11380
IT staff                 526
Laborers               55186
Low-skill Laborers      2093
Managers               21371
Medicine staff          8537
Private service staff   2652
Realty agents            751
Sales staff            32102
Secretaries             1305
Security staff          6721
Waiters/barmen staff    1348
dtype: int64
```

In [26]:
```
fig = px.histogram(ap_train, width=800, height=500,x="OCCUPATION_TYPE", color="TARGET", barmode="group", log_y=True, template="simple_white", co
fig.update_layout(xaxis=dict(showgrid=False, ), yaxis=dict(showgrid=False))
fig.show()
```



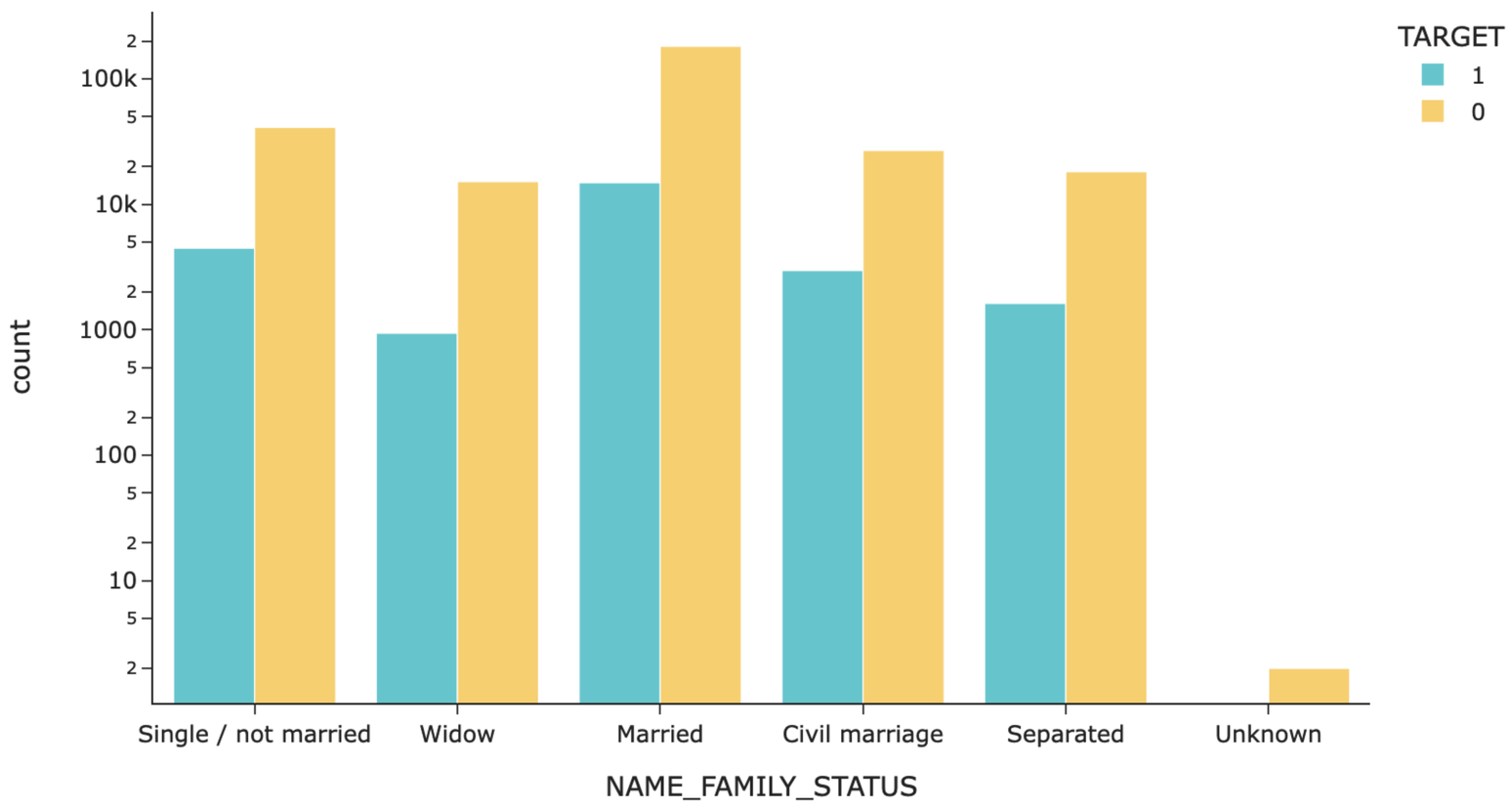
The variable indicates what kind of occupation the client has at the moment of applying for the loan. It reports that there are 96391 missing values which do not correspond to those who had declared to be unemployed because the number was significantly lower (22 clients). So, one should reflect on why 31% of clients preferred to not state their work. This could be related to the fact that there is no category in which clients can express "other" types of work or it could be due to the fact that clients do not feel comfortable expressing their job. Concerning the target variable, all the occupation categories report at least 82% of solvent clients, so there is no evidence that belonging to a certain job category affects the client's solvency.

NAME_FAMILY_TYPE

In [27]:
```
ap_train.groupby('NAME_FAMILY_STATUS').apply(len)
```

Out[27]:
```
NAME_FAMILY_STATUS
Civil marriage          29775
Married                196432
Separated               19770
Single / not married    45444
Unknown                     2
Widow                   16088
dtype: int64
```

In [28]:
```
fig = px.histogram(ap_train,width=800, height=500, color="TARGET", x="NAME_FAMILY_STATUS", barmode="group", log_y=True, template="simple_white",
fig.update_layout(xaxis=dict(showgrid=False, ), yaxis=dict(showgrid=False))
fig.show()
```



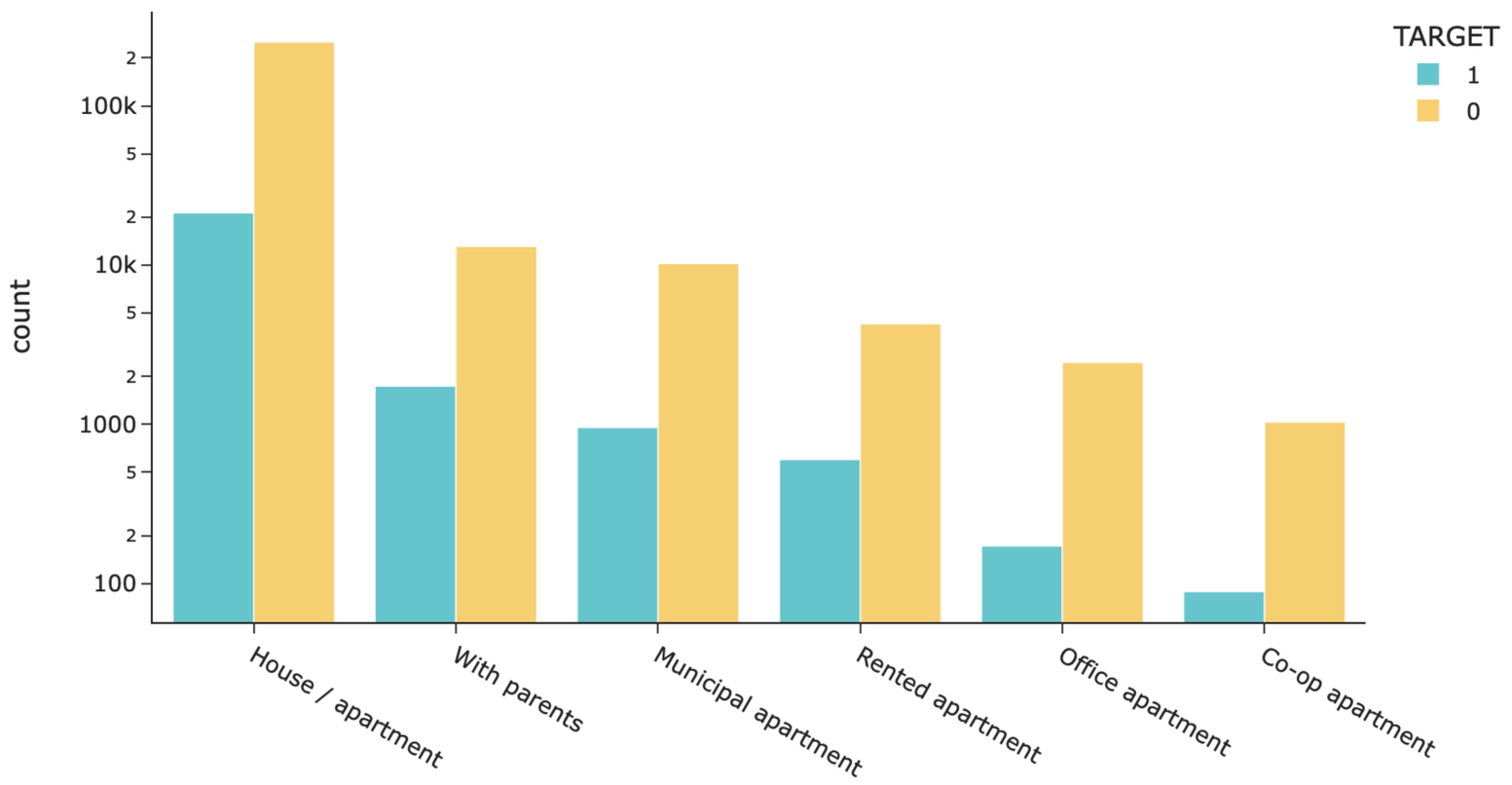
The variable represents the family status of the client. It must be considered that there are 2 different categories for people married: one is related to civil marriage, so the other will probably refer to a religious marriage. Belong to the latter category 63% of clients. Regarding the relationship with the target variable, at least 90% in each category has no payment difficulties, so it is not possible to say that this variable indicates the client's solvency.

NAME_HOUSING_TYPE

In [29]:
```
ap_train.groupby('NAME_HOUSING_TYPE').apply(len)
```

Out[29]:
```
NAME_HOUSING_TYPE
Co-op apartment       1122
House / apartment    272868
Municipal apartment   11183
Office apartment       2617
Rented apartment       4881
With parents          14840
dtype: int64
```

In [30]:
```
fig = px.histogram(ap_train,width=800, height=500, color="TARGET", x="NAME_HOUSING_TYPE", barmode="group", log_y=True, template="simple_white",
fig.update_layout(xaxis=dict(showgrid=False, ), yaxis=dict(showgrid=False))
fig.show()
```



The variable indicates what is the housing situation of the client:

- House/apartment
- Rented apartment
- With parents
- Municipal apartment
- Office apartment
- Co-op apartment

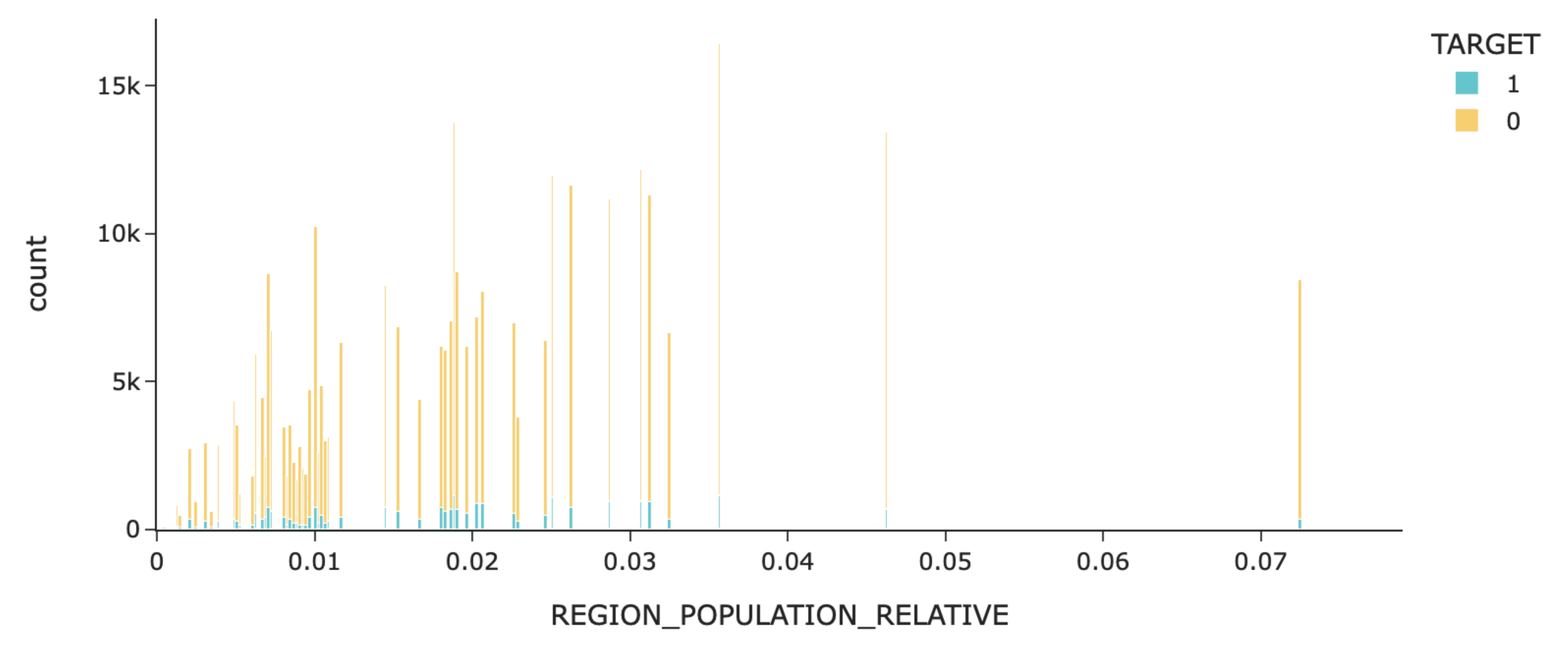
With respect to the target variable, the graph shows that 88% of clients results to have a house or apartment, and among these only 8% have some payment difficulties. The 12% of clients who have rented apartments and the 11% who live with parents have some payment difficulties, but they are only a few cases over the overall number of clients.

REGION_POPULATION_RELATIVE

In [31]:
```
ap_train['REGION_POPULATION_RELATIVE'].describe()
```

Out[31]:
```
count    307511.000000
mean          0.020868
std           0.013831
min           0.000290
25%           0.010006
50%           0.018850
75%           0.028663
max           0.072508
Name: REGION_POPULATION_RELATIVE, dtype: float64
```

In [32]:
```
fig = px.histogram(ap_train,width=800, height=400, x="REGION_POPULATION_RELATIVE", color="TARGET", range_x=[0,0.079],template="simple_white", co
fig.update_layout(xaxis=dict(showgrid=False, ), yaxis=dict(showgrid=False))
fig.show()
```



The variable represents a normalisation of the regional population in which the customer lives. It can take values between 0.000290 and 0.072508: a higher number means that the customer lives in a more populated region. The graph shows that most customers live in sparsely populated regions.

DAYS_BIRTH

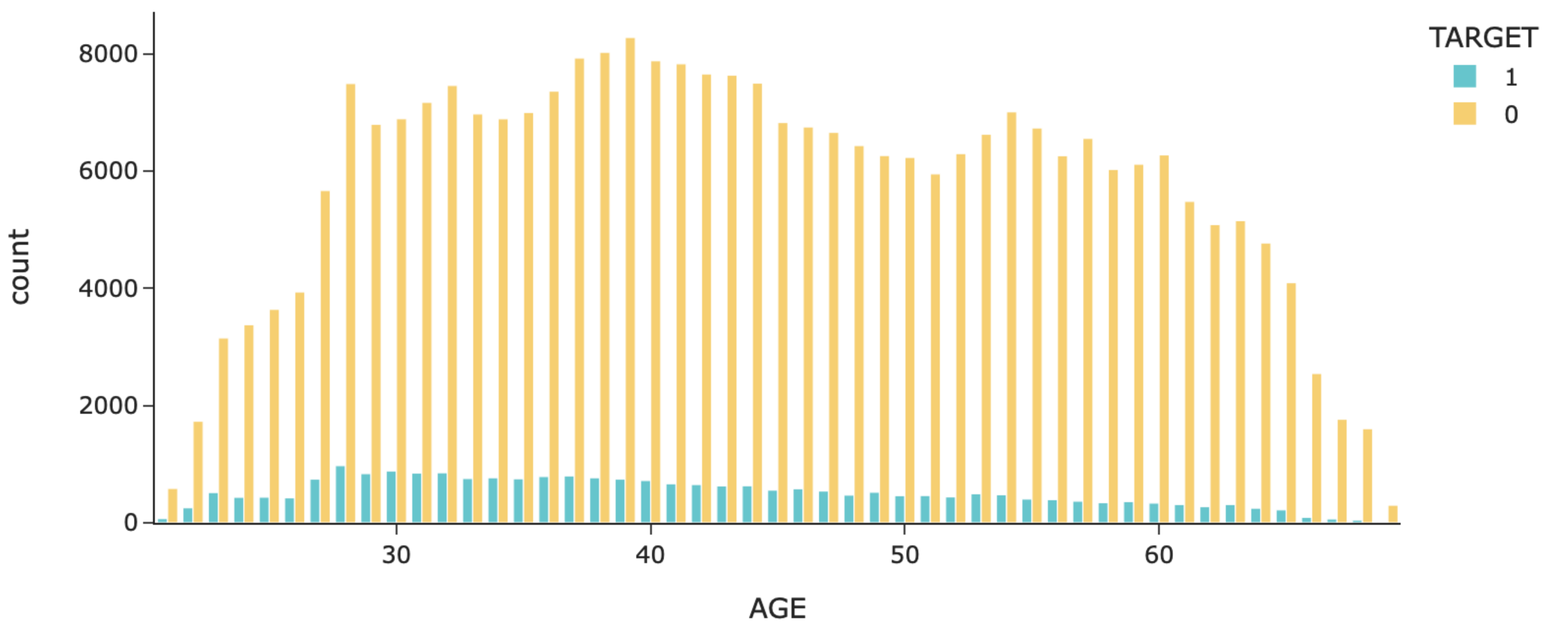
In [33]:
```
ap_train['AGE'].min()
```

Out[33]:
```
21.0
```

In [34]:
```
ap_train['AGE'].max()
```

Out[34]:
```
69.0
```

In [35]:
```
fig = px.histogram(ap_train, width=800, height=400, x="AGE", color="TARGET", barmode="group",template="simple_white", color_discrete_sequence=px
fig.update_layout(xaxis=dict(showgrid=False, ), yaxis=dict(showgrid=False))
fig.show()
```



The variable reports the age of clients which range goes from 21 to 69. Overall, the variable has no relevant implication on the target, but, as expected, the younger age group reflects a more difficult payment behaviour.

DAYS_EMPLOYED

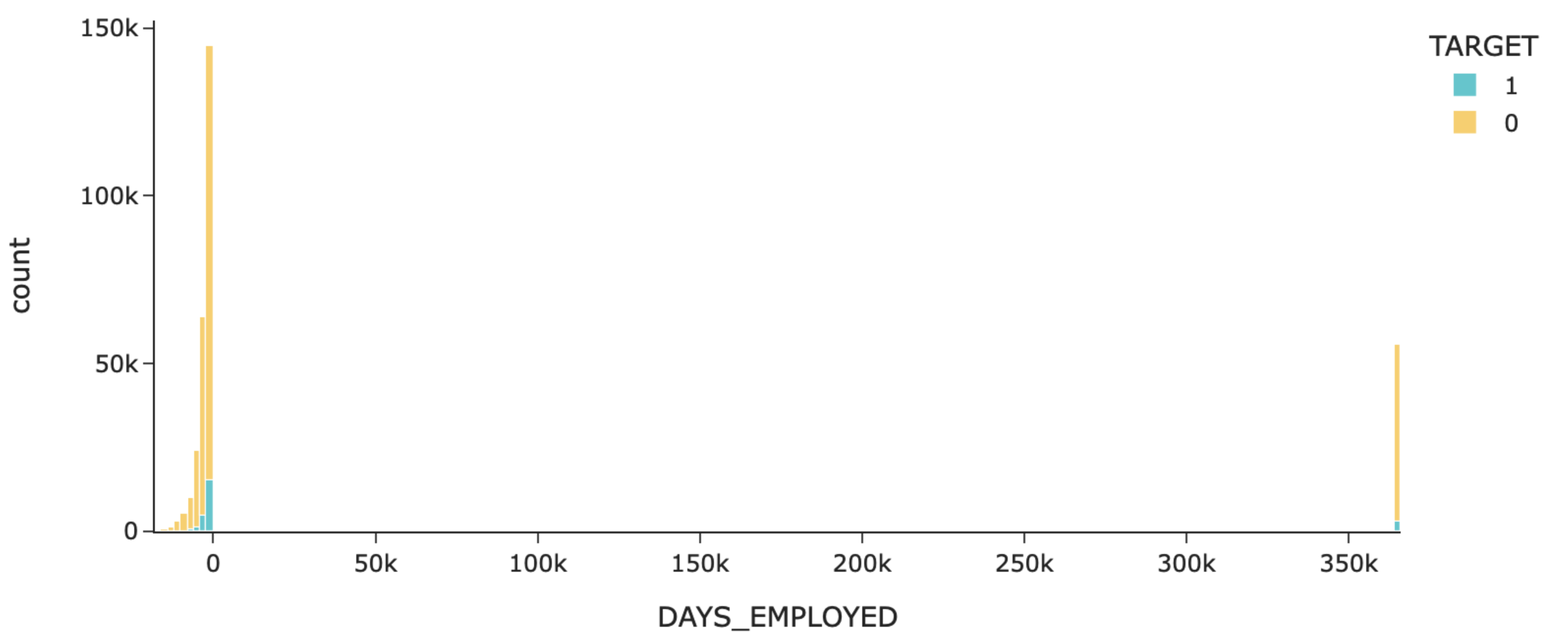
In [36]:
```
ap_train['DAYS_EMPLOYED']
```

Out[36]:
```
0           -637
1          -1188
2           -225
3          -3039
4          -3038
           ...
307506      -236
307507    365243
307508     -7921
307509     -4786
307510     -1262
Name: DAYS_EMPLOYED, Length: 307511, dtype: int64
```

In [37]:
```
year_employed = ap_train['DAYS_EMPLOYED']//365
year_employed.describe()
```

Out[37]:
```
count    307511.000000
mean        174.303742
std         386.994347
min         -50.000000
25%          -8.000000
50%          -4.000000
75%          -1.000000
max        1000.000000
Name: DAYS_EMPLOYED, dtype: float64
```

In [38]:
```
fig = px.histogram(ap_train, width=800, height=400, x="DAYS_EMPLOYED", color="TARGET",template="simple_white", color_discrete_sequence=px.colors
fig.update_layout(xaxis=dict(showgrid=False, ), yaxis=dict(showgrid=False))
fig.show()
```



The variable describes how many days before the loan application the person started their current employment. As already analyzed most of the clients are employed. However, it should be noted that there are a lot of negative values of which we do not know the correct interpretation and then the values go up to more than 350k days. The interpretation is not straightforward, even trying to transform days into years the results do not change so much: the minimum value is always negative (-50) and the maximum is very high (1000) so it doesn't reflect the reality.

DAYS_ID_PUBLISH

In [39]:
```
ap_train.groupby('DAYS_ID_PUBLISH').apply(len)
```

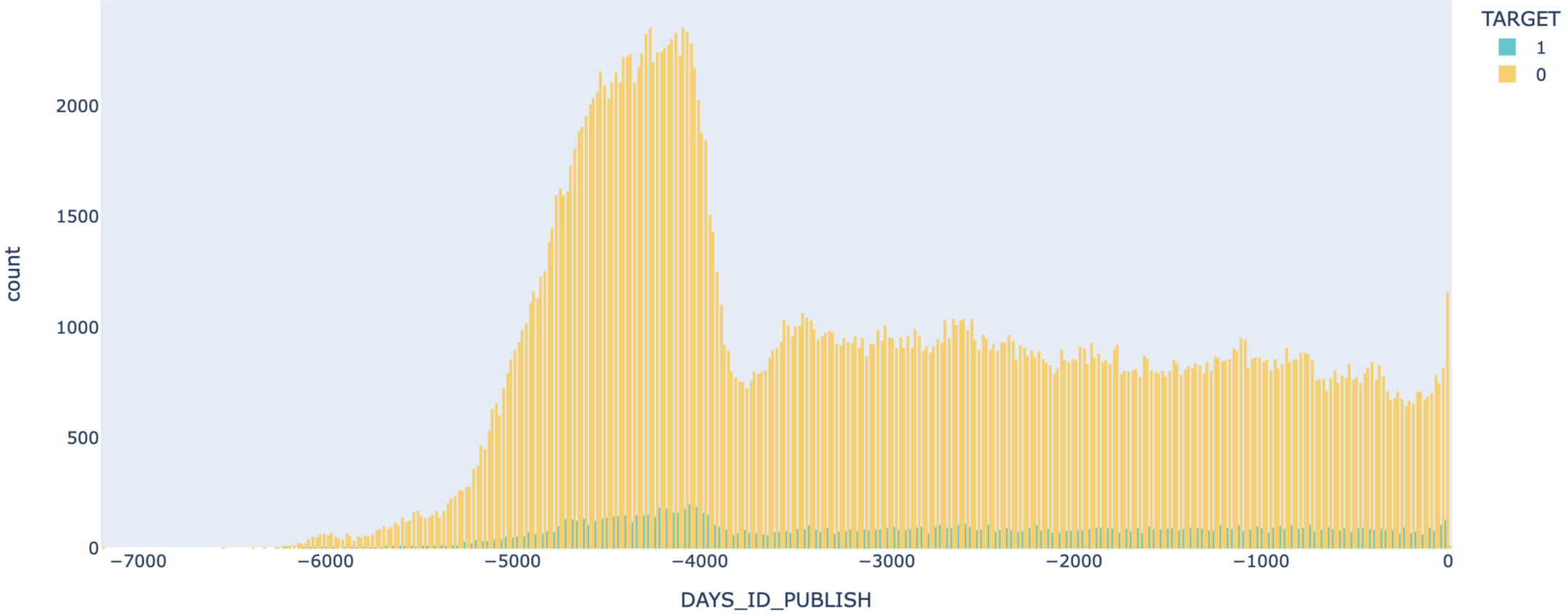
Out[39]:
```
DAYS_ID_PUBLISH
-7197     1
-6551     1
-6383     1
-6337     1
-6274     1
         ..
-4       57
-3       51
-2       50
-1       64
```

```
 0       16
Length: 6168, dtype: int64
```

In [40]:
```
year_employed = ap_train['DAYS_ID_PUBLISH']//365
year_employed.describe()
```

Out[40]:
```
count    307511.000000
mean         -8.710732
std           4.134511
min         -20.000000
25%         -12.000000
50%          -9.000000
75%          -5.000000
max           0.000000
Name: DAYS_ID_PUBLISH, dtype: float64
```

In [42]:
```
fig = px.histogram(ap_train, x="DAYS_ID_PUBLISH", color="TARGET", barmode="group", color_discrete_sequence=px.colors.qualitative.Pastel)
fig.update_layout(xaxis=dict(showgrid=False, ), yaxis=dict(showgrid=False))
fig.show()
```



The variable indicates how many days before the application did client change the identity document with which he applied for the loan. The variable interpretation is not straightforward because there are most negative values. By transforming the days in years, it is possible to interpret that the maximum number of years before did client change the ID is 20 which is weird because documents are usually updated more frequently in reality.

OWN_CAR_AGE

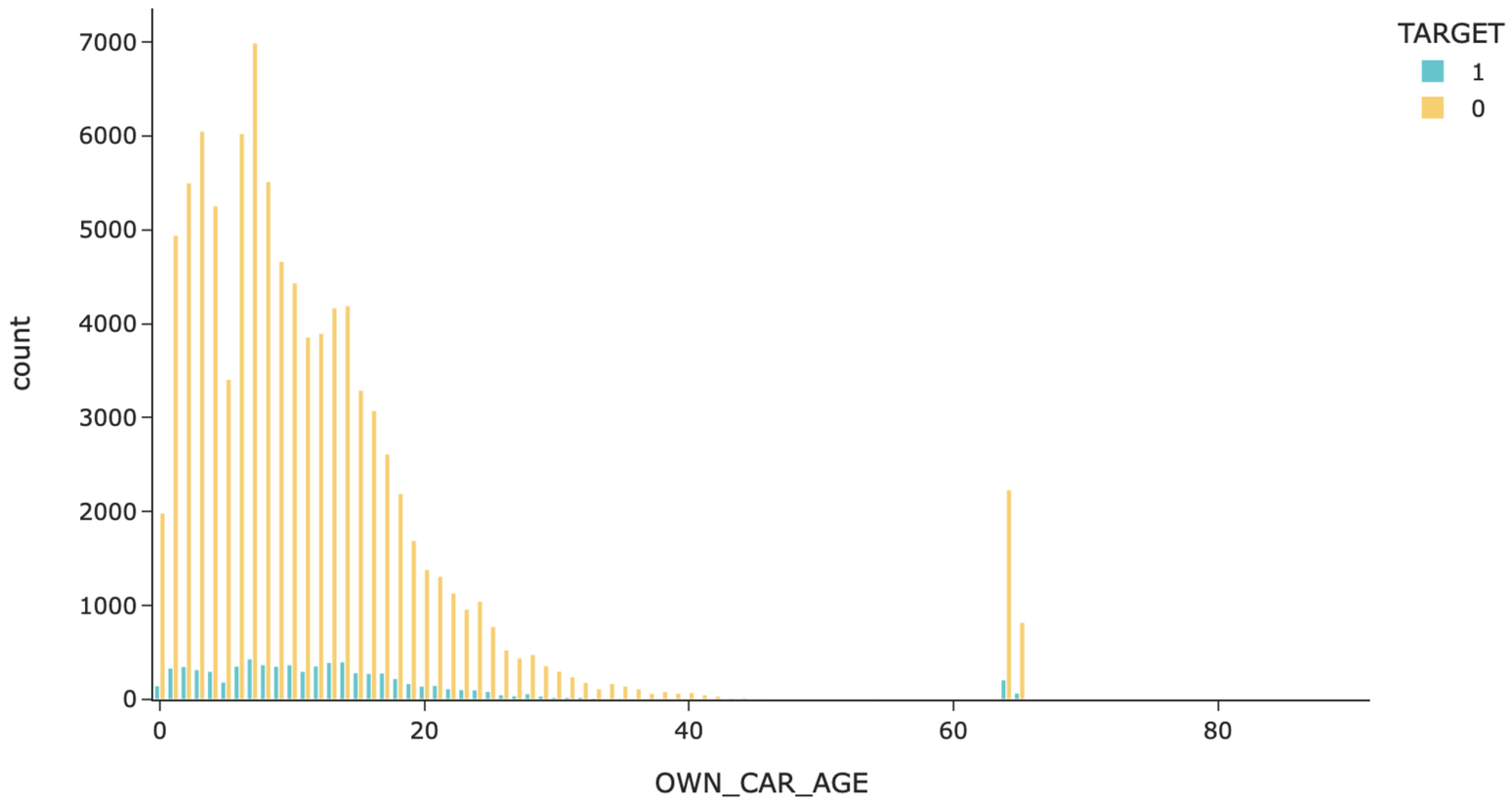
In [43]:
```
ap_train['OWN_CAR_AGE'].isnull().sum()
```

Out[43]:
```
202929
```

In [44]:
```
ap_train.groupby('OWN_CAR_AGE').apply(len)
```

Out[44]:
```
OWN_CAR_AGE
0.0     2134
1.0     5280
2.0     5852
3.0     6370
4.0     5557
        ...
63.0       2
64.0    2443
65.0     891
69.0       1
91.0       2
Length: 62, dtype: int64
```

In [45]:
```
fig = px.histogram(ap_train, width=800, height=500,x="OWN_CAR_AGE", color="TARGET", barmode="group",template="simple_white", color_discrete_sequ
fig.update_layout(xaxis=dict(showgrid=False, ), yaxis=dict(showgrid=False))
fig.show()
```



The variable indicates the age of the client's car. The car age range goes from 0 to 91. Further, 65% of the values (202929) are missing. This is coherent with what was discovered in the analysis of clients who own a car, in which the results show that 202924 clients do not have a car. So, there are only 5 missing values related to the car age. With respect to the target variable, it is not a meaningful variable because an old-age car can be vintage or second-hand.

Preparing the data

In order to prepare data for an ML model, the first step is the encoding of features.

In [46]:
```
ap_train['CODE_GENDER_M'] = np.select([ap_train['CODE_GENDER'] == 'M', ap_train['CODE_GENDER'] == 'F'], [1, 0], default=np.NaN)
ap_train['FLAG_OWN_CAR'] = np.where(ap_train['FLAG_OWN_CAR'] == 'Y', 1, 0)
ap_train['FLAG_OWN_REALTY'] = np.where(ap_train['FLAG_OWN_REALTY'] == 'Y', 1, 0)
ap_train.drop(columns='CODE_GENDER', inplace=True)
```

In [47]:
```
ap_objects = list(ap_train.select_dtypes(include=['object']).columns)
ap_train[ap_objects] = ap_train[ap_objects].astype('category')
```

Separate the target from the rest of the data.

In [48]:
```
ap_train_target = ap_train.pop('TARGET')
print(f"Target dataset shape: {ap_train_target.shape}")
```

```
Target dataset shape: (307511,)
```

Split the original dataset into 2 datasets:

- 80% for the train set
- 20% for the validation set, which will correspond to the test set of the project

In [49]:
```
df_train, df_test, df_target_train, df_target_test = train_test_split(
    ap_train, ap_train_target, test_size=0.2, stratify=ap_train_target, random_state=42)

print(f"Train dataset shape: {df_train.shape}")
print(f"Test dataset shape: {df_test.shape}")
```

```
Train dataset shape: (246008, 25)
Test dataset shape: (61503, 25)
```

Create a basic ML model and scoring on the test set

The model performed is a XGBoost whose parameters were chosen beforehand by cross-validation.

In [50]:
```
df_train_dmatrix = xgb.DMatrix(df_train.drop(columns='SK_ID_CURR'), df_target_train, enable_categorical=True)

param = {'max_depth': 6,
         'eta': .2,
         'subsample': .9,
         'colsample_bytree': .9,
         'scale_pos_weight': 10,
         'objective': 'binary:logistic',
         'tree_method': 'exact'}

xgb_base_model = xgb.train(param, df_train_dmatrix, num_boost_round=50)
```

Score the test set:

In [51]:
```
df_test_dmatrix = xgb.DMatrix(df_test.drop(columns='SK_ID_CURR'), enable_categorical=True)
xgb_base_test_results = xgb_base_model.predict(df_test_dmatrix)
```

- What are the risk scores (from 0 to 1) of the first 5 customers in the test set? And what's the overall AUC on the test set?

In [52]:
```
df_test.iloc[:5,:]
```

Out[52]:

	SK_ID_CURR	NAME_CONTRACT_TYPE	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRICE
256571	396899	Cash loans	1	1	1	157500.0	770292.0	30676.5	688500.0
191493	322041	Cash loans	0	0	0	90000.0	364896.0	19926.0	315000.0
103497	220127	Cash loans	0	1	0	148500.0	284400.0	18643.5	225000.0
130646	251531	Cash loans	0	0	0	188100.0	976711.5	38218.5	873000.0
211898	345558	Cash loans	0	1	0	180000.0	323194.5	19660.5	279000.0

5 rows × 25 columns

In [53]:
```
xgb_base_test_results[:5]
```

Out[53]:
```
array([0.45560062, 0.23749042, 0.7490776 , 0.24548666, 0.49250585],
      dtype=float32)
```

The risk scores are between the 0.24 and 0.75 for the first 5 customers in the test set.

In [54]:
```
fpr, tpr, thresholds = metrics.roc_curve(df_target_test, xgb_base_test_results, pos_label=1)
benchm_auc = metrics.auc(fpr, tpr)
print(benchm_auc)
```

```
0.7541543982734993
```

0.75 is the probability that the classifier correctly ranks a random client among those who are solvent or who have payment difficulties.

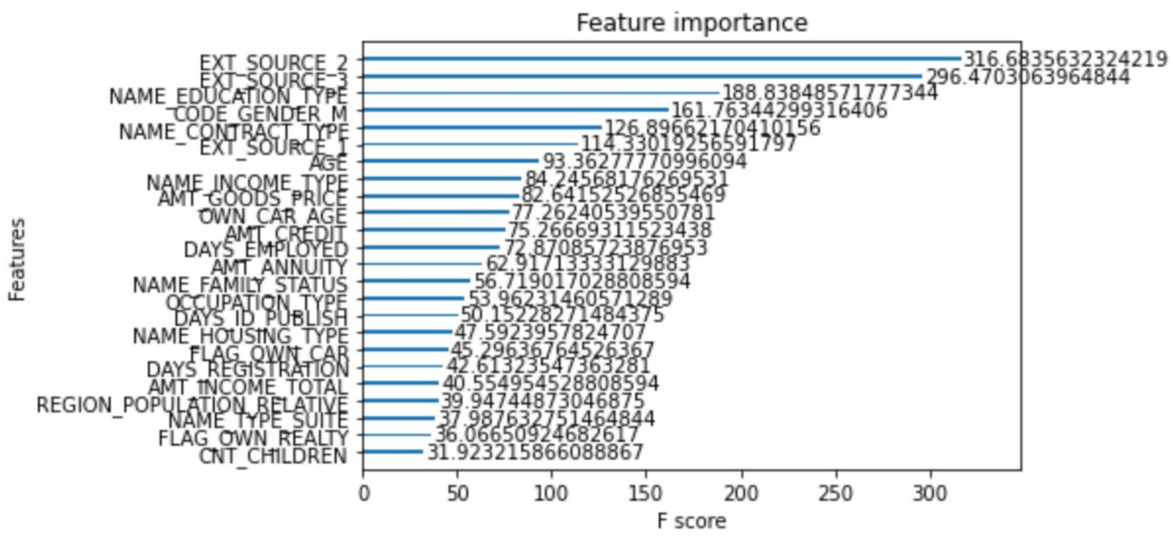
[#3] Evaluating feature importance

The third step is to understand whether the ethical variables highlited are actually important in the runned model

[#3.1] Evaluating feature importance using built-in xgboost plot_importance

In [55]:
```
xgb.plot_importance(xgb_base_model, importance_type='gain', grid=False, max_num_features = 25)

# The Gain is the most relevant attribute to interpret the relative importance of each feature.
# 'Gain' is the improvement in accuracy brought by a feature to the branches it is on.
# The idea is that before adding a new split on a feature X to the branch there was some wrongly classified elements,
# after adding the split on this feature, there are two new branches, and each of these branch is more accurate.
```

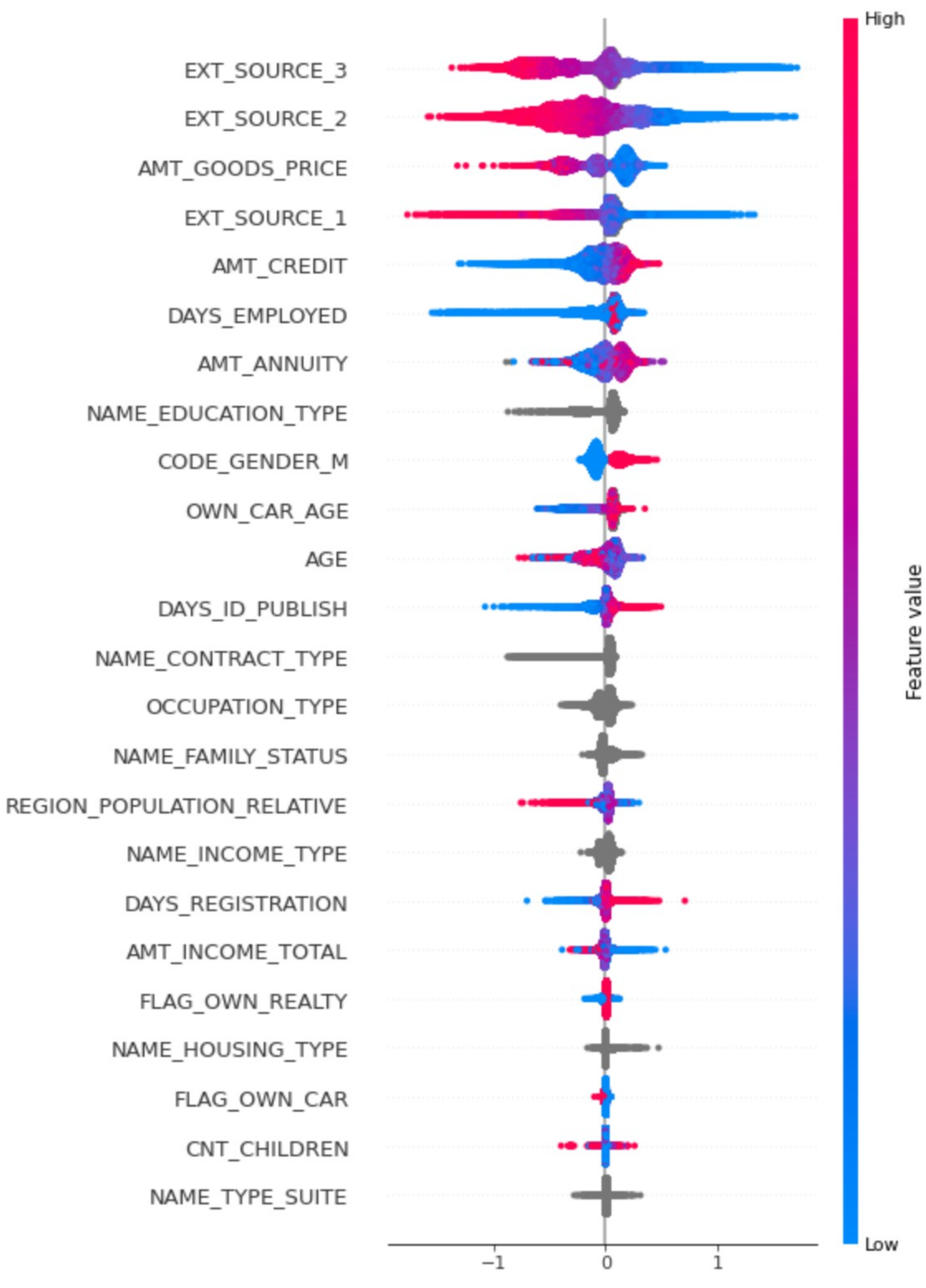
Out[55]:
```
<AxesSubplot:title={'center':'Feature importance'}, xlabel='F score', ylabel='Features'>
```



[#3.2] Evaluating feature importance using SHAP plot_importance

Interpretation Documentation

In [56]:
```
def shap_eval():
    max_disp = df_test.shape[1]
    explainer = shap.TreeExplainer(xgb_base_model)
    shap_values = explainer.shap_values(df_test_dmatrix)
    shap.summary_plot(shap_values, df_test.iloc[:,1:], max_display = max_disp)
shap_eval()
```



SHAP value (impact on model output)

The feature importance of the variables is computed with 2 techniques:

- the standard built-in feature importance focus on the relation between the variable and the prediction. The importance is calculated with the importance_type that in this case is Gain, which is the average gain of splits that use the feature,
- the SHAP values which calculate the importance of a feature by comparing what a model predicts with and without the feature and since the order in which a model sees features can affect its predictions, this is done in every possible order, so that the features are fairly compared. From the SHAP values on the x-axis, it is possible to extract the probability of success. Whereas the gradient color indicates the original value for that variable, for example, it is grey for categorical variables, such as in the case of the education level, it is 2 colours whether the variable is binary, such as in the case of gender. The scatter points don't fit on a line they pile up to show density, and the color of each point represents the feature value of that individual. SHAP values come with consistency guarantees in terms of ordering the features correctly.

The approach to this task is to take into consideration both models. They highlighted that the 2 most relevant variables are external sources (EXT_SOURCE_3 and EXT_SOURCE_2) whose information is not known, but it is possible to say that they have more total model impact than the EXT_SOURCE_1 that for those samples where it matters, it has more impact than the amount of CREDIT. On the other hand, one of the least relevant variables seems to be the number of children the loan applicant has.

From the SHAP, it is interesting to note that, in addition to the external sources, also the variables related to the loan (such as annuity, and the amount of credit) have more total model impact than the ethical features. Concerning the latter, those that appear to have an impact on the model output are:

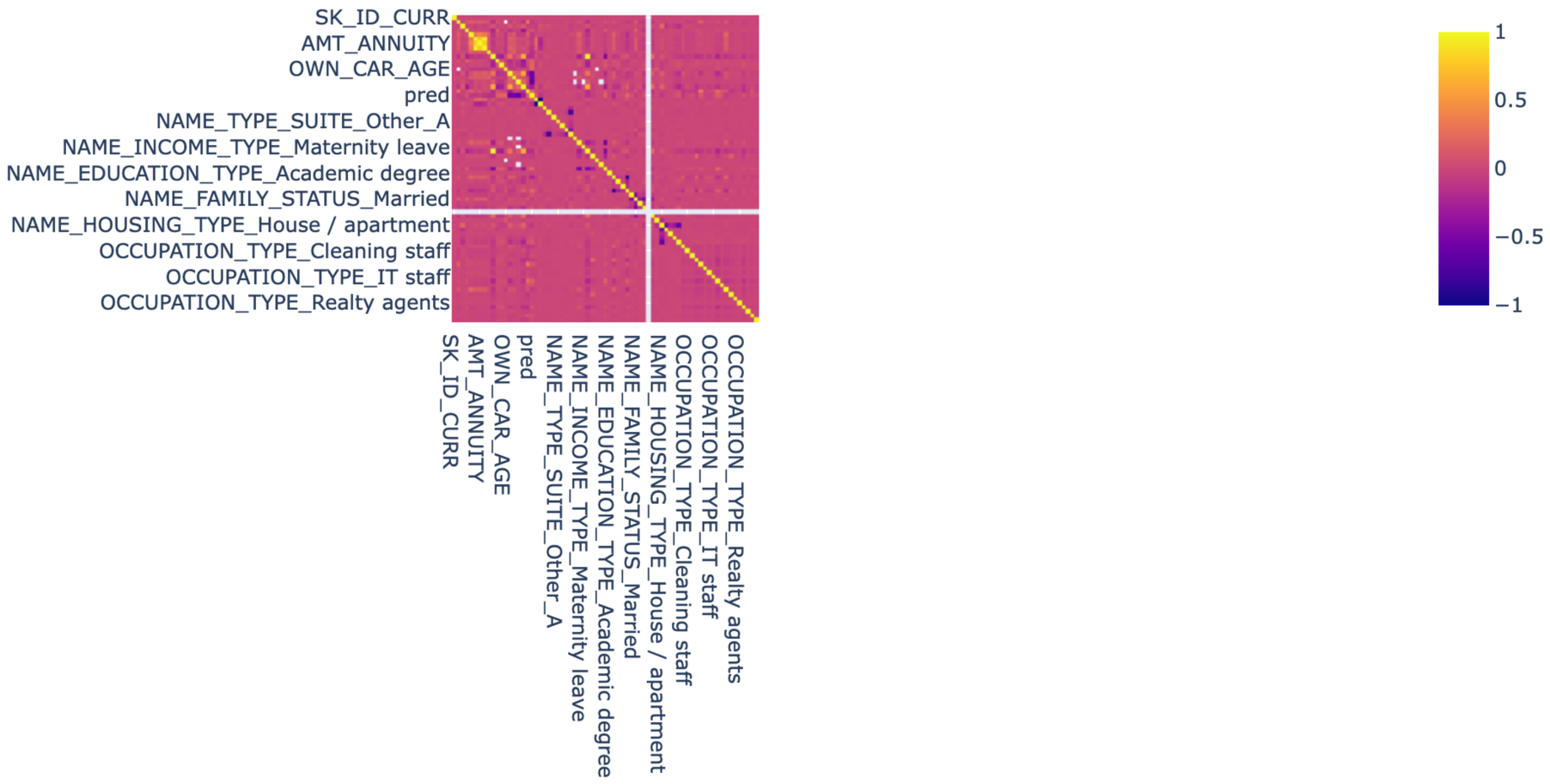
- the level of education affects all predictions by a small amount,
- the gender of the clients affects few predictions by a large amount (this is the case of men) and more predictions by a smaller amount (in the case of women),
- the age of the client affects all predictions by a small amount,
- the number of days before the loan request the client started current employment effect few predictions, whether the age is high by a smaller amount than whether it is low by a large amount.
- the number of days before the loan request the client change is ID affects few predictions by a large amount.
- region population relative. In this case, there are few regions with a high-level population concentration affected by a large amount, and most low-populated regions are effected by a smaller amount.

[#4] Comparing the model predictions with respect to the original training data

The fourth step concerns the analysis of the realtion between the predictions and the ethical features with respect to what was bring out in the second task.

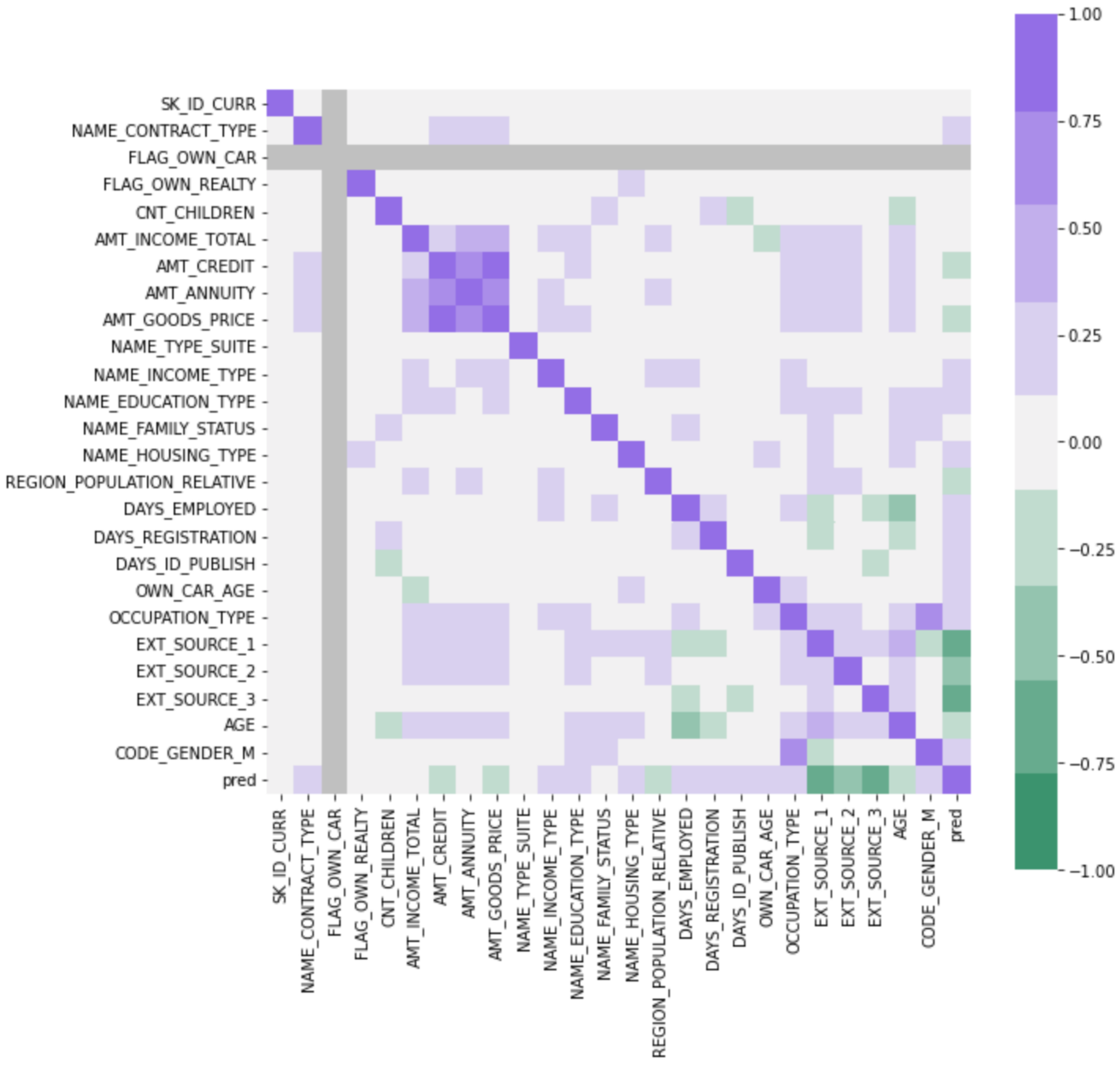
[#4.1] Manual correlation matrix after one hot encoding

In [57]:
```
df_test2 = df_test.copy()
df_test2.reset_index(drop = True, inplace = True)

xgb_base_test_results = pd.DataFrame(xgb_base_test_results, columns = ['pred'])

test_correlation = pd.concat([df_test2, xgb_base_test_results],
                             axis = 1)

test_correlation_dummies = pd.get_dummies(test_correlation)

fig = px.imshow(test_correlation_dummies.corr())
fig.show()
```



[#4.2] Correlation matrix using Dython

In [58]:
```
complete_correlation = associations(test_correlation, figsize=(10,10), nan_strategy = 'drop_samples', vmin=-1, vmax=1,
                                    cmap=sns.diverging_palette(150, 275, s=80, l=55, n=9), annot= False)
```



With regard to the ethical variables highlighted in the previous tasks, the overall behaviour between the test characteristics and the test predictions provided by the ML model remained largely unchanged from that found previously. This is consistent with what was expected, as the model has a relatively high AUC (0.75) and thus most of the test predictions are correctly classified among the actual payment behaviour.

By looking at the correlation matrixes, the negative correlation between the predicted values and the external sources is stronger than the one originally seen.

Regarding the ethical variables, stand out that there is a slight positive correlation (0.28) between the predictions and the education level variable, which was not so evident before. And the same is for the days before the application the person started the current employment and the occupation type (respectively 0.29 and 0.28). Moreover, the code gender begins to have more weight on predictions (0.19)

Other slight positive correlations erase with the income type, the type of house, the number of days before the clients change ID, and the own car age (respectively 0.14, 0.12, 0.12, 0.18).

In addition, some slight negative correlations erase with the age and region population relative variables (respectively -0.27 and -0.17).

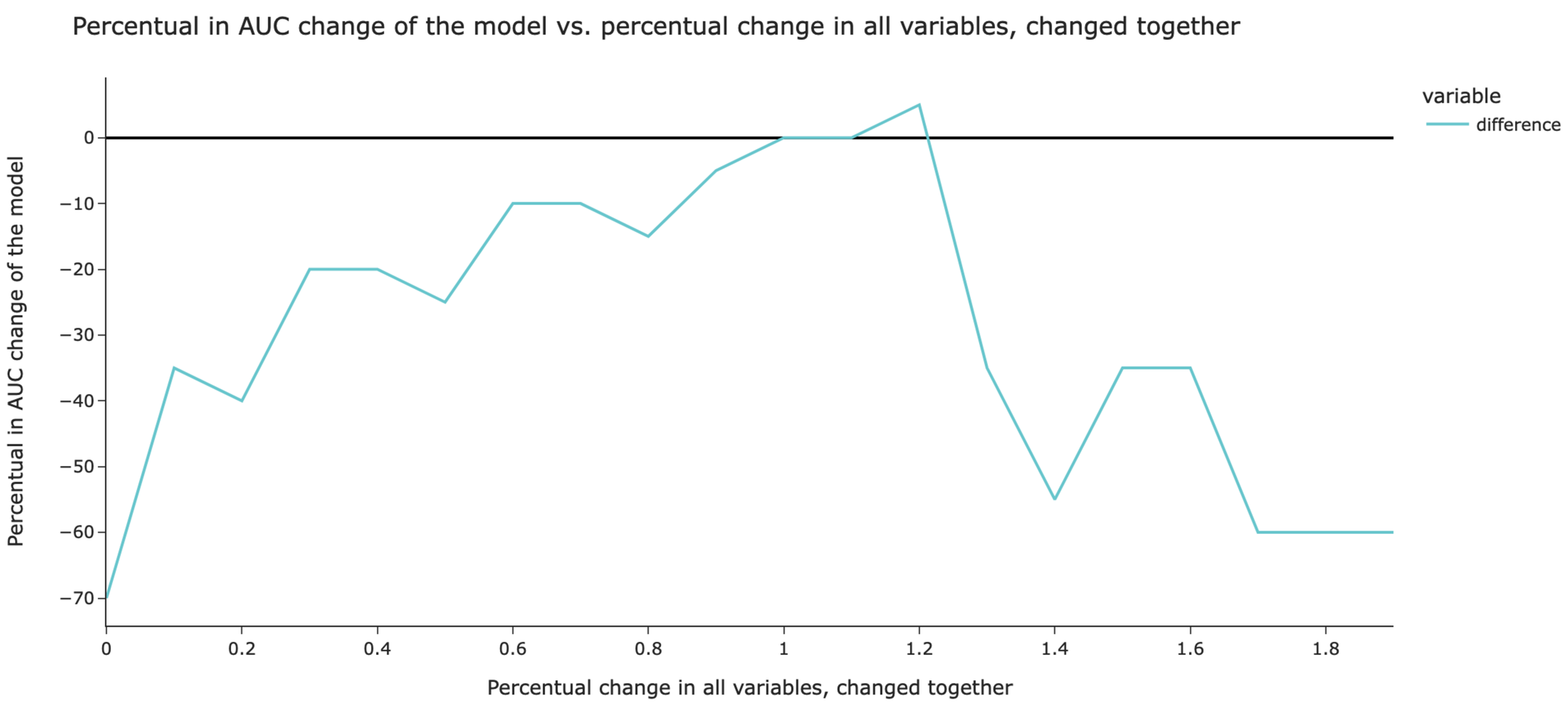
[#5] Alter ethical variables

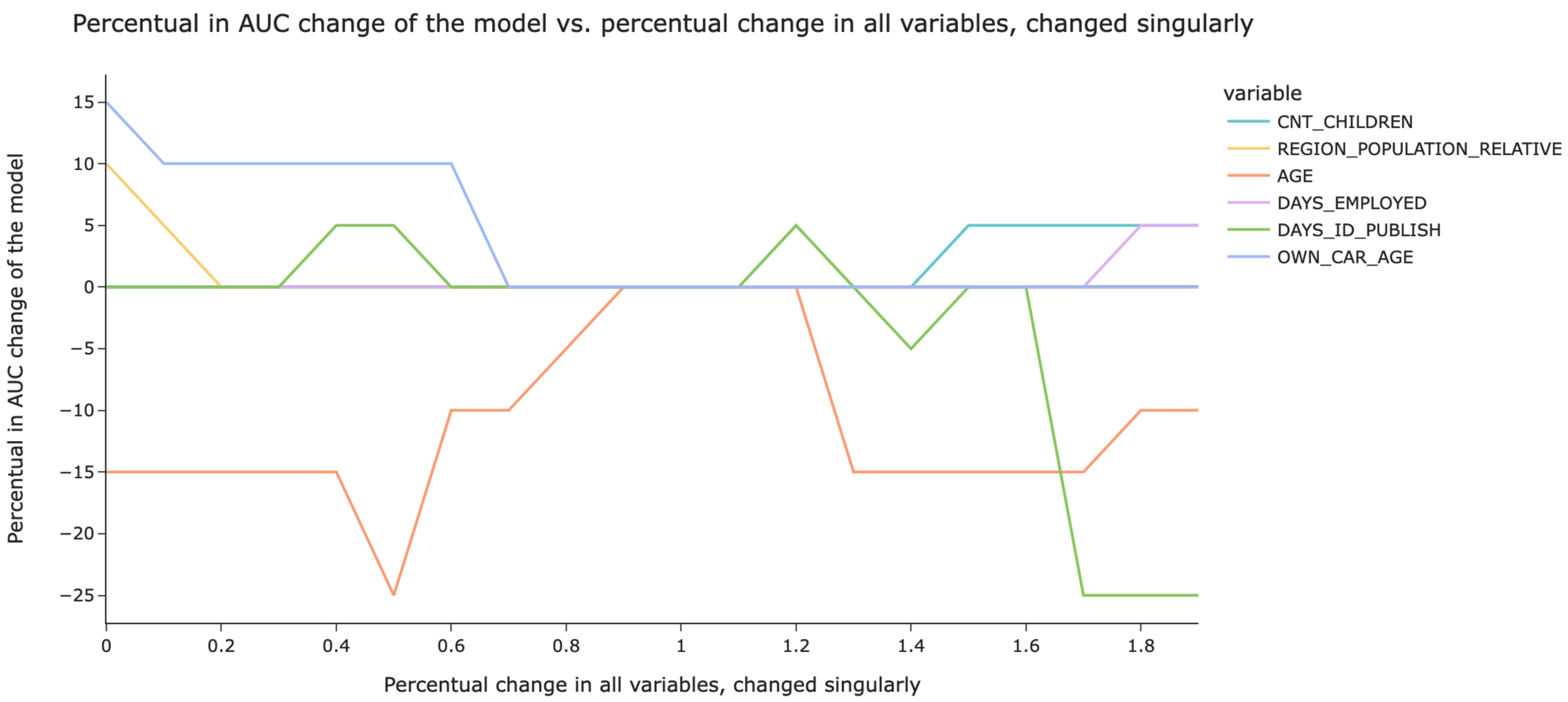
The fifth step consist of manually altering some records and find out how model performance changes. The alteration is first computed on the numerical variables, altered all together and individually, then on the categorical variables altered individually and all together.

Numerical variables:

In [59]:
```
variables = ['CNT_CHILDREN', 'REGION_POPULATION_RELATIVE' , 'AGE', 'DAYS_EMPLOYED',
             'DAYS_ID_PUBLISH', 'OWN_CAR_AGE']

def plot_difference(option):
    # option argument could be "singularly" or "together"

    # Defining the "fitting" function, which will repeat in any cases and return the results dictionary
    def fitting():

        # Fit the model and predict
        df_test_dmatrix_mod = xgb.DMatrix(df_test_mod.drop(columns='SK_ID_CURR'),
                                          enable_categorical=True)

        xgb_base_test_results_mod = xgb_base_model.predict(df_test_dmatrix_mod)

        # Store the percentual change in the results dictionary and transform it to a dataframe
        results[change] = metrics.roc_auc_score(df_target_test[:number], xgb_base_test_results_mod)

        results[change] = ((results[change] - benchmark) / benchmark) * 100

        return results

    # Define the number of samples taken as subset of test set
    number = 30

    # Calculate the benchmark performance
    benchmark = metrics.roc_auc_score(df_target_test[:number], xgb_base_test_results[:number])

    if option == 'together':

        # Create a dictionary to store the changes
        results = dict()

        # Iterate over 0 to 100%, with 10% steps
        for change in np.arange(0, 2, 0.1):

            # Take a subset of the test set
            df_test_mod = df_test.iloc[:number, :].copy(deep=True)
```

```
            # For each column in the dataset
            for var in df_test_mod:
                # If the column is in the variables we chose:
                if var in variables:
                    # Change the column by the given change
                    df_test_mod[var] = df_test_mod[var] * change

            results = fitting()

        df = pd.DataFrame.from_dict(results, orient = 'index', columns = ['difference'])

    elif option == 'singularly':

        # Define an empty dataset
        df = pd.DataFrame()

        # Iterate over each variable
        for var in variables:

            # Stores the results in a dictionary
            results = dict()

            #For each number in [0%;200%] with 10% step
            for change in np.arange(0, 2, 0.1):

                # Get the copy of the dataset with a length == number
                df_test_mod = df_test.iloc[:number,:].copy(deep = True)

                # Change the column by the given step
                df_test_mod[var] = df_test_mod[var] * change

                results = fitting()

            # Create a temporary dataframe, which has as index the steps and the column is the variable name; the cell is the relative change in
            temp_df = pd.DataFrame.from_dict(results, orient = 'index', columns = [var])

            # Concatenate the dataframe
            df = pd.concat([df, temp_df], axis = 1)

    # Plot the resulting dataframe
    fig = px.line(df,
                  x = df.index,
                  y = df.columns[:],
                  template="simple_white",
                  color_discrete_sequence=px.colors.qualitative.Pastel)

    fig.update_xaxes(zeroline=True, zerolinewidth=2, zerolinecolor='Black')
    fig.update_yaxes(zeroline=True, zerolinewidth=2, zerolinecolor='Black')
    fig.update_layout(paper_bgcolor='rgba(0,0,0,0)',plot_bgcolor='rgba(0,0,0,0)')
    fig.update_layout(title="Percentual in AUC change of the model vs. percentual change in all variables, changed {}".format(option),
                      xaxis_title="Percentual change in all variables, changed {}".format(option),
                      yaxis_title="Percentual in AUC change of the model")

    fig.show()


plot_difference('together')
plot_difference('singularly')
```





Altering the numerical variables, the output consists of 2 graphs.

The first graph describes how much the AUC changes when there is a certain percentual variation in all the features together. The point 1 on the x-axis represents the situation in which the features do not change, indeed the change in the AUC is equal to 0. As it can be seen, whether all the variables change by +30%, the AUC decreases a lot, almost reduced by half, and so the model is no longer accurate.

The second graph describes how much the AUC changes and whether there is a certain percentual variation in each variable individually at different times. As can be seen, the AUC is very affected (up to -25%) by a low value of clients' age and high values of days regarding the ID change. Whereas, regarding the number of children and the number of days before the client started the job, it remains mostly stable with exception of high values of these variables which lead to an increase in the AUC. With respect to the variables about the own car age and the region population relative, as they decrease by more than 50%, the AUC increases respectively by 15% and 10% and that is coherent with what is highlighted with the correlation matrix.

Categorical variables

In [60]:
```
df_test['TARGET'] = df_target_test
```

In [61]:
```
#BENCHMARK AUC
df_test_altered = df_test.sample(n=5, random_state = 3).reset_index(drop=True)

#Compute the new AUC
df_target_altered_test = df_test_altered.pop('TARGET')
df_test_altered_dmatrix = xgb.DMatrix(df_test_altered.drop(columns='SK_ID_CURR'), enable_categorical=True)
xgb_base_test_results = xgb_base_model.predict(df_test_altered_dmatrix)
#fpr, tpr, thresholds = metrics.roc_curve(df_target_altered_test, xgb_base_test_results, pos_label=1)
#print(xgb_base_test_results)
auc_altered = metrics.roc_auc_score(df_target_altered_test, xgb_base_test_results)
print(auc_altered)
```

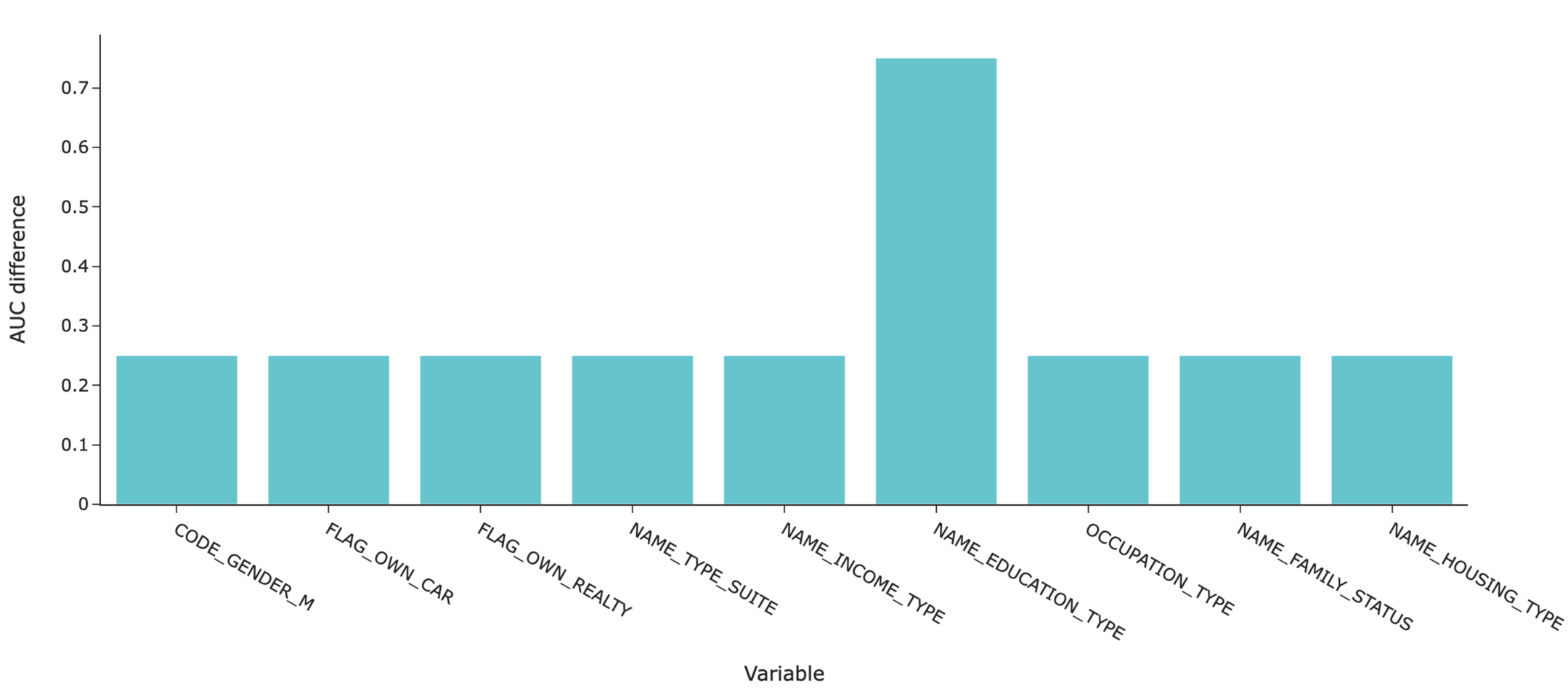
```
0.5
```

In [62]:
```
auc_differences = {"variable":[],"difference":[]};

def AUCdifference(column: str):
    #Get unique values of the column
    unique_vals = list(df_test[column].dropna().unique())
    #Get a subset of size 5
    df_test_altered = df_test.sample(n=5, random_state = 3).reset_index(drop=True)

    #Change values of the column by randomly taking elements from the unique values list
    np.random.seed(23)
    index = -1
    for el in df_test_altered[column]:
        index = index + 1
        df_test_altered.at[index,column]=np.random.choice(unique_vals)

    #Compute the new AUC
    df_target_altered_test = df_test_altered.pop('TARGET')
    df_test_altered_dmatrix = xgb.DMatrix(df_test_altered.drop(columns='SK_ID_CURR'), enable_categorical=True)
    xgb_base_test_results = xgb_base_model.predict(df_test_altered_dmatrix)
    fpr, tpr, thresholds = metrics.roc_curve(df_target_altered_test, xgb_base_test_results, pos_label=1)
    #print(xgb_base_test_results)
    auc_altered = metrics.roc_auc_score(df_target_altered_test, xgb_base_test_results)
    #print(auc_altered)

    auc_difference = 0.75-auc_altered

    #Append the AUC difference to the dictionary
    auc_differences["variable"].append(column)
    auc_differences["difference"].append(auc_difference)

    print('The AUC difference when the variable ' + str(column) + ' is altered is: ' + str(auc_difference))
```

In [63]:
```
AUCdifference('CODE_GENDER_M')
AUCdifference('FLAG_OWN_CAR')
AUCdifference('FLAG_OWN_REALTY')
AUCdifference('NAME_TYPE_SUITE')
AUCdifference('NAME_INCOME_TYPE')
AUCdifference('NAME_EDUCATION_TYPE')
AUCdifference('OCCUPATION_TYPE')
AUCdifference('NAME_FAMILY_STATUS')
AUCdifference('NAME_HOUSING_TYPE')
```

```
The AUC difference when the variable CODE_GENDER_M is altered is: 0.25
The AUC difference when the variable FLAG_OWN_CAR is altered is: 0.25
The AUC difference when the variable FLAG_OWN_REALTY is altered is: 0.25
The AUC difference when the variable NAME_TYPE_SUITE is altered is: 0.25
The AUC difference when the variable NAME_INCOME_TYPE is altered is: 0.25
The AUC difference when the variable NAME_EDUCATION_TYPE is altered is: 0.75
The AUC difference when the variable OCCUPATION_TYPE is altered is: 0.25
The AUC difference when the variable NAME_FAMILY_STATUS is altered is: 0.25
The AUC difference when the variable NAME_HOUSING_TYPE is altered is: 0.25
```

In [64]:
```
fig = px.bar(auc_differences, x='variable', y='difference', template="simple_white", color_discrete_sequence=px.colors.qualitative.Pastel)
fig.update_layout(title="AUC change when categorical ethical variables are singularly changed",
                  xaxis_title="Variable",
                  yaxis_title="AUC difference")
fig.update_layout(paper_bgcolor='rgba(0,0,0,0)',plot_bgcolor='rgba(0,0,0,0)')
fig.update_layout(xaxis=dict(showgrid=False, ), yaxis=dict(showgrid=False))
fig.show()
```



For categorical variables, the alteration is computed individually by changing the values of the column by randomly taking elements from the unique values list of each column for 5 records. The AUC of these few observations is now equal to 0.5 but we consider as a benchmark the original one which is equal to 0.75.

As changing the values, the AUC difference with the benchmark is about 0.25, which means that the AUC with the altered variable decreases to 0.50. The only exception is the education level which involves a difference of 0.75, so by altering this variable the AUC of the model decreases to 0.

- Should it, therefore, be considered a relevant variable with respect to a person's payment behaviour?

In [65]:
```
#Initial altered dataset
df_test_altered = df_test.sample(n=5, random_state = 3).reset_index(drop=True)
df_test_altered
```

Out[65]:

	SK_ID_CURR	NAME_CONTRACT_TYPE	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRICE	NAME
0	373797	Cash loans	0	1	0	90000.0	900000.0	31887.0	900000.0	U
1	400654	Cash loans	1	0	1	157500.0	545040.0	20677.5	450000.0	U
2	161462	Cash loans	1	1	0	112500.0	675000.0	24376.5	675000.0	
3	155514	Cash loans	0	1	0	135000.0	305221.5	19503.0	252000.0	S
4	167047	Cash loans	1	1	0	112500.0	360000.0	19530.0	360000.0	U

5 rows × 26 columns

In [66]:
```
#Alter all the categorical variables together
categorical = ['CODE_GENDER_M', 'FLAG_OWN_CAR', 'FLAG_OWN_REALTY',
               'NAME_TYPE_SUITE', 'NAME_INCOME_TYPE', 'NAME_EDUCATION_TYPE',
               'OCCUPATION_TYPE', 'NAME_FAMILY_STATUS', 'NAME_HOUSING_TYPE']

df_test_altered = df_test.sample(n=5, random_state = 3).reset_index(drop=True)

np.random.seed(44)
for col in df_test_altered.columns:
    index = -1
    if col in categorical:
        unique_vals = list(df_test[col].dropna().unique())
        for el in df_test_altered[col]:
            index = index + 1
            df_test_altered.at[index,col]=np.random.choice(unique_vals)
```

In [67]:
```
#Dataset after the alteration of all categorical variables
df_test_altered
```

Out[67]:

	SK_ID_CURR	NAME_CONTRACT_TYPE	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRICE	NAME
0	373797	Cash loans	1	0	0	90000.0	900000.0	31887.0	900000.0	S
1	400654	Cash loans	0	1	1	157500.0	545040.0	20677.5	450000.0	
2	161462	Cash loans	0	1	0	112500.0	675000.0	24376.5	675000.0	G
3	155514	Cash loans	0	0	0	135000.0	305221.5	19503.0	252000.0	U
4	167047	Cash loans	0	0	0	112500.0	360000.0	19530.0	360000.0	

5 rows × 26 columns

In [68]:
```
#Compute the new AUC
df_target_altered_test = df_test_altered.pop('TARGET')
df_test_altered_dmatrix = xgb.DMatrix(df_test_altered.drop(columns='SK_ID_CURR'), enable_categorical=True)
xgb_base_test_results = xgb_base_model.predict(df_test_altered_dmatrix)
fpr, tpr, thresholds = metrics.roc_curve(df_target_altered_test, xgb_base_test_results, pos_label=1)
print(xgb_base_test_results)
auc_altered = metrics.roc_auc_score(df_target_altered_test, xgb_base_test_results)
print(auc_altered)

print('The difference with the bechmark AUC is of ' + str(0.75-auc_altered))
```

```
[0.3739864  0.37177536 0.5805099  0.2415329  0.08984957]
0.0
The difference with the bechmark AUC is of 0.75
```

Trying to alter all the categorical variables together leads to a drastic decrease in the AUC values which becomes equal to 0 so the predictions would be totally wrong to such an extent that the labels are switched.

[#6] Did the model learn the differences and biases in the original dataset?

The sixth step regards an overall analysis about the model

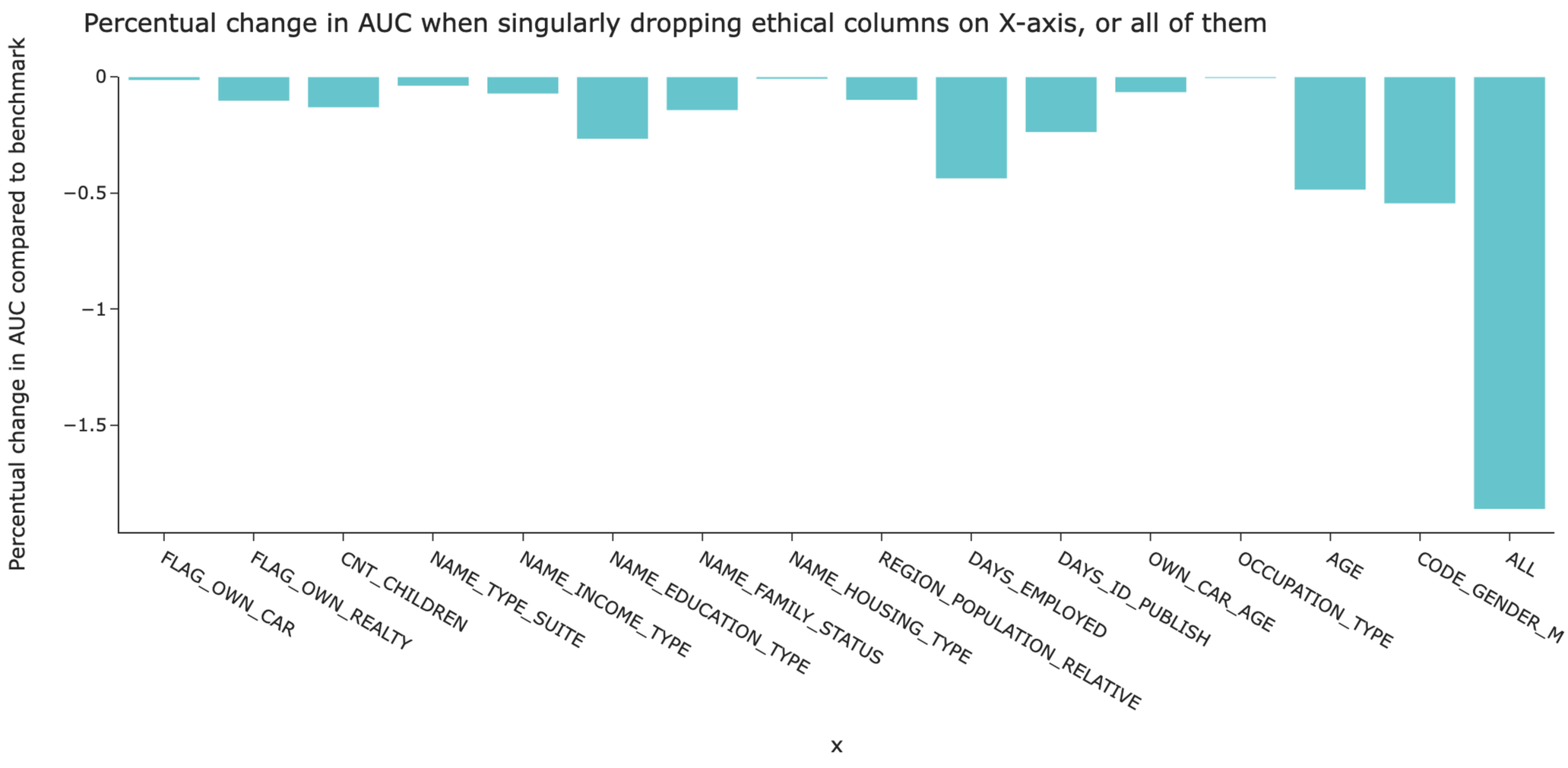
The overall behaviour of the model on the test set was already highlighted in the correlation matrix between the predictions and all the other features. It is coherent with what is find out by altering the ethical variable of some record. It is evident that numerical variables influence the AUC more than categorical ones. Furthermore, by altering all the ethical variables together the AUC changes and this is what one would expect regardless of the relevant ethics. The bias in the model still results in the case of single alteration of ethical variables because altering the value of a single ethical variable should not weigh so heavily on the model, as is the case for the variable of age or level of education.

[#7] Remove ethical variables and compute the new AUC performance

The seventh step consists of removing the ethical variables from the model and studying what happens

In [69]:
```
# Let's list the variable we considered ethical relevant
ethic_var = ['FLAG_OWN_CAR', 'FLAG_OWN_REALTY', 'CNT_CHILDREN',
             'NAME_TYPE_SUITE', 'NAME_INCOME_TYPE', 'NAME_EDUCATION_TYPE',
             'OCCUPATION_TYPE', 'NAME_FAMILY_STATUS', 'NAME_HOUSING_TYPE',
             'REGION_POPULATION_RELATIVE', 'DAYS_EMPLOYED', 'OWN_CAR_AGE',
             'AGE', 'DAYS_ID_PUBLISH', 'CODE_GENDER_M']

def removing_ethics():

    # Enclose the preprocessing lines from the first blocks in the preprocessing function
    def preprocessing():
        ap_train = pd.read_csv('application_train.csv',
                        usecols =  ['TARGET', 'SK_ID_CURR', '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', 'OCCUPATION_TYPE',
                                    'NAME_FAMILY_STATUS', 'NAME_HOUSING_TYPE', 'REGION_POPULATION_RELATIVE', 'DAYS_BIRTH',
                                    'DAYS_EMPLOYED', 'DAYS_REGISTRATION', 'DAYS_ID_PUBLISH', 'OWN_CAR_AGE', 'EXT_SOURCE_1',
                                    'EXT_SOURCE_2', 'EXT_SOURCE_3'])
        ap_train['AGE'] = -round(ap_train['DAYS_BIRTH'] / 365)
        ap_train.drop(columns='DAYS_BIRTH', inplace=True)
        ap_train['CODE_GENDER_M'] = np.select([ap_train['CODE_GENDER'] == 'M', ap_train['CODE_GENDER'] == 'F'], [1, 0], default=np.NaN)
        ap_train['FLAG_OWN_CAR'] = np.where(ap_train['FLAG_OWN_CAR'] == 'Y', 1, 0)
        ap_train['FLAG_OWN_REALTY'] = np.where(ap_train['FLAG_OWN_REALTY'] == 'Y', 1, 0)
        ap_train.drop(columns='CODE_GENDER', inplace=True)
        return ap_train

    # Same with processing lines and fitting, return the auc metrics
    def processing_and_fitting(x):
        ap_objects = list(x.select_dtypes(include=['object']).columns)
        x[ap_objects] = x[ap_objects].astype('category')
        y = x.pop('TARGET')
        df_train, df_test, df_target_train, df_target_test = train_test_split(
            x, y, test_size=0.2, stratify=ap_train_target, random_state=42)
        df_train_dmatrix = xgb.DMatrix(df_train.drop(columns='SK_ID_CURR'), df_target_train, enable_categorical=True)
        param = {'max_depth': 6,
                 'eta': .2,
                 'subsample': .9,
                 'colsample_bytree': .9,
                 'scale_pos_weight': 10,
                 'objective': 'binary:logistic',
                 'tree_method': 'exact'}
        xgb_base_model = xgb.train(param, df_train_dmatrix, num_boost_round=50)
        df_test_dmatrix = xgb.DMatrix(df_test.drop(columns='SK_ID_CURR'), enable_categorical=True)
        xgb_base_test_results = xgb_base_model.predict(df_test_dmatrix)
        fpr, tpr, thresholds = metrics.roc_curve(df_target_test, xgb_base_test_results, pos_label=1)

        return metrics.auc(fpr, tpr)

    # Create a new dictionary to store auc percentual difference in metric, every time you remove an ethical column
    results = dict()
    dfz = preprocessing()

    # Iterate each column in ap_train
    for col in dfz:
        dfz = preprocessing()

        # If the column is ethical, drop it and at each iteration measure the percentual difference in metrics between new metric and the benchm
        if col in ethic_var:
            dfz.drop([col], axis=1, inplace = True)
            results[col] = ((processing_and_fitting(dfz) - benchm_auc) / benchm_auc) * 100

    # Convert the result dictionary into a dataframe and plot the differences
    dfg = pd.DataFrame.from_dict(results, orient = 'index', columns=['difference'])

    # Repeat the process, adding to the dataframe related to the difference in performance when you remove all the columns at the same time
    all_df = preprocessing()
    for col in all_df:
```

```
        if col in ethic_var:
            all_df.drop([col], axis=1, inplace = True)

    dfg = pd.concat([dfg, pd.DataFrame.from_dict({'ALL': ((processing_and_fitting(all_df) - benchm_auc) / benchm_auc) * 100},
                                                 orient  = 'index', columns = ['difference'])], axis = 0)

    # Plot the resulting dataframe
    fig = px.bar(x=dfg.index, y=dfg.difference, template="simple_white",color_discrete_sequence=px.colors.qualitative.Pastel)
    fig.update_layout(paper_bgcolor='rgba(0,0,0,0)',plot_bgcolor='rgba(0,0,0,0)')
    fig.update_layout(xaxis=dict(showgrid=False, ), yaxis=dict(showgrid=False))
    fig.update_layout(title="Percentual change in AUC when singularly dropping ethical columns on X-axis, or all of them",
                      yaxis_title="Percentual change in AUC compared to benchmark", template="simple_white")

    fig.show()

removing_ethics()
```



Removing all the ethical variables implies a change in the model AUC metric. The graph above shows how much the AUC changes when the ethical variables are dropping singularly and the last bin represents the case when all of them are dropped together.

As already mentioned, there are some categorical variables that weigh more heavily on the model due to their relevance to the target variable, but in general, dropping the variable results in a negative change in the AUC of at most 0.55 %. In particular, the variables that result more important whether dropped are:

- the gender
- the age
- the number of days before the application for the loan, the client started the current employment
- the education level

When all ethical variables were removed, the percentage of change in AUC decreased by - 1.86%.

[#8] Do we still see differences for the average predicition of different groups?

The eighth step regards the discussion of a model withouth any ethical variable.

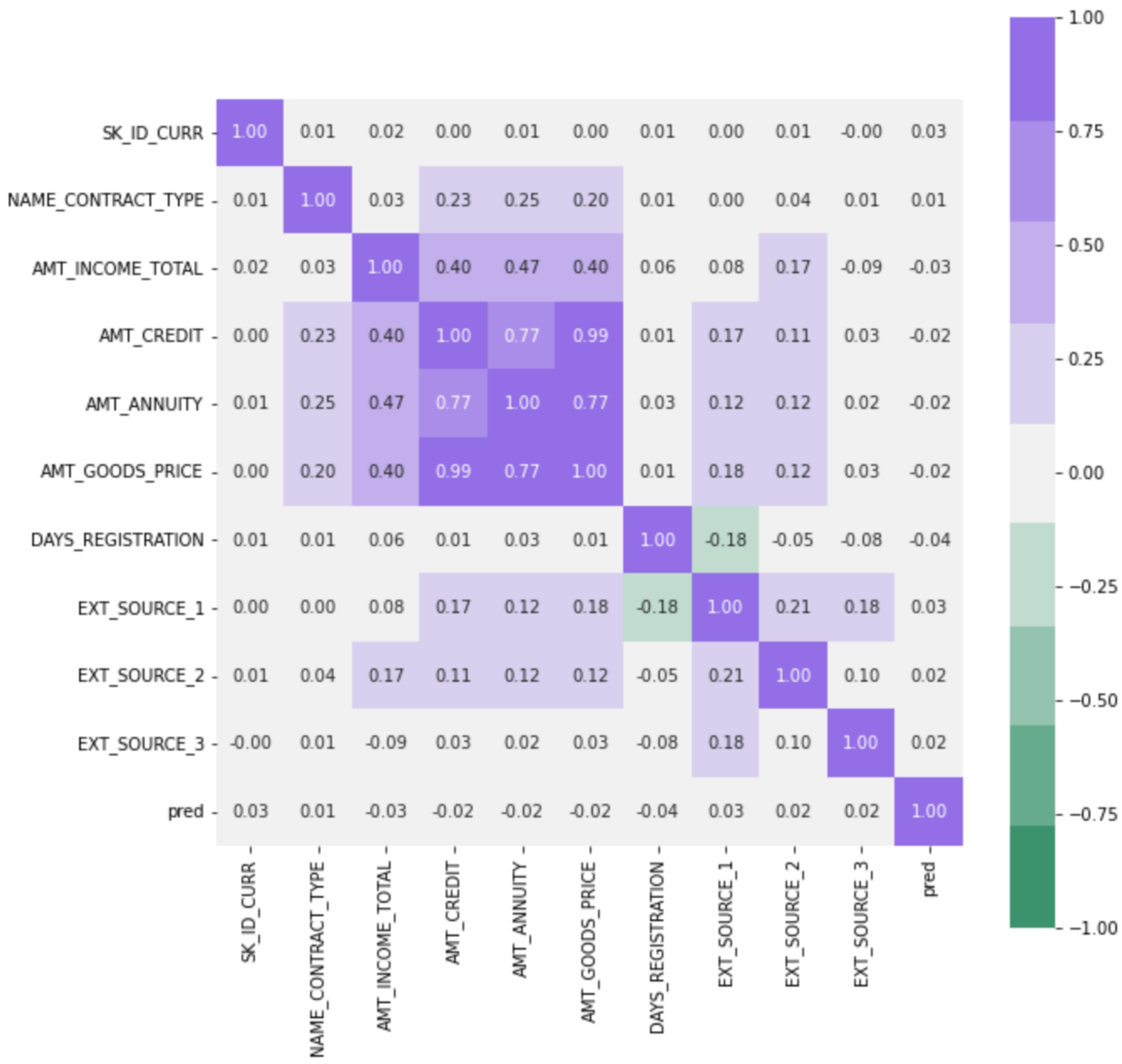
In [70]:
```
train_noethics=df_train.copy()
train_noethics.drop(columns=['CODE_GENDER_M', 'FLAG_OWN_CAR', 'FLAG_OWN_REALTY', 'CNT_CHILDREN',
             'NAME_TYPE_SUITE', 'NAME_INCOME_TYPE', 'NAME_EDUCATION_TYPE',
             'OCCUPATION_TYPE', 'NAME_FAMILY_STATUS', 'NAME_HOUSING_TYPE',
             'REGION_POPULATION_RELATIVE', 'DAYS_EMPLOYED', 'OWN_CAR_AGE',
             'AGE', 'DAYS_ID_PUBLISH'], inplace=True)
test_noethics=df_test.copy()
test_noethics.drop(columns=['CODE_GENDER_M', 'FLAG_OWN_CAR', 'FLAG_OWN_REALTY', 'CNT_CHILDREN',
             'NAME_TYPE_SUITE', 'NAME_INCOME_TYPE', 'NAME_EDUCATION_TYPE',
             'OCCUPATION_TYPE', 'NAME_FAMILY_STATUS', 'NAME_HOUSING_TYPE',
             'REGION_POPULATION_RELATIVE', 'DAYS_EMPLOYED', 'OWN_CAR_AGE',
             'AGE', 'DAYS_ID_PUBLISH', 'TARGET'], inplace=True)
```

In [71]:
```
train_noethics_dmatrix = xgb.DMatrix(train_noethics.drop(columns='SK_ID_CURR'), df_target_train, enable_categorical=True)

param = {'max_depth': 6,
         'eta': .2,
         'subsample': .9,
         'colsample_bytree': .9,
         'scale_pos_weight': 10,
         'objective': 'binary:logistic',
         'tree_method': 'exact'}

xgb_base_model_noethics = xgb.train(param, train_noethics_dmatrix, num_boost_round=50)
test_noethics_dmatrix = xgb.DMatrix(test_noethics.drop(columns='SK_ID_CURR'), enable_categorical=True)
xgb_base_test_results_noethics = xgb_base_model_noethics.predict(test_noethics_dmatrix)
fpr, tpr, thresholds = metrics.roc_curve(df_target_test, xgb_base_test_results_noethics , pos_label=1)
auc = metrics.auc(fpr, tpr)
print(auc)
```

```
0.7401154681518374
```

It is therefore confirmed that eliminating all ethical variables causes a slight decrease in AUC which is not as significant as expected.

In [72]:
```
xgb_base_test_results_noethics = pd.DataFrame(xgb_base_test_results_noethics, columns = ['pred'])

test_correlation = pd.concat([test_noethics, xgb_base_test_results_noethics],
                             axis = 1)
complete_correlation = associations(test_correlation, figsize=(10,10), nan_strategy = 'drop_samples', vmin=-1, vmax=1,
                                    cmap=sns.diverging_palette(150, 275, s=80, l=55, n=9), annot= True)
```



After removing the ethical variable the relationship between the variables and the predictions undergoes some changes. The target variable appears to be not correlated with any of the other dataset's variables, not even more with the external sources, which were originally more negatively correlated. At this point, at this point, it seems that all variables are equally slightly correlated to the predictions.

[#9] Can you explain why just removing the variables wasn't enough?

The ninth step regards an in-depth understanding of how model performs.

Previous results were not as expected, it seems that dropping the ethical variable does not affect the performance of the model.Whereas, one would have expected that dropping the 15 variables the AUC decreases significantly, because it is thought that if these variables are originally included in a model, it is because they have a bearing on the payment behaviour. It must therefore be noted what was find out in the previous tasks:

- at the beginning, The relationship between the target and the variables does not show that there are any subcategories of variables that can be associated with payment behaviour, whether positive or negative. This sounded good because the clients cannot be categorized as solvent or not based on ethical features.
- whether the ethical variables are altered individually, the change in the AUC change depends on the weight of that variable in the model.
- whether the ethical variables are altered all at once, the AUC can change considerably. As seen with the alteration of all the categorical ethical variables in which case the AUC
- whether the ethical variables are dropped individually , the AUC decrease is not so significant, at most the -0.54 of the benchmark AUC.
- whether the ethical variables are dropped all together, the AUC decreases but not by as much as expected.

This final result could be due to the specific case because, in theory, one would expect a strong trade off between bias and accuracy. Furthermore, the number of features has been greatly reduced in order to be able to interpret the results, and this can certainly lead to a model that is detached from reality.

Then, in order to understand the problem in depht, it is necessary to abstract from the specific case and the specific AUC values. Even because at the beginning of the analysis, the classes of the different ethical variables seemed to have no particular connection to the target variable, whereas after running the model, it extrapolated some correlations. As is known, machine learning algorithm can determine new relationships that a person would never think to test. Whether these relationships have casual properties or are only proxies for other correlated factors are critical questions in determining the legality and ethics of using machine learning in many fields, including the financial services sector. As in this case, where it is necessary to analyse potential benefits and risks of entrusting the choice of granting a loan to an AI model.

The ethical biases are embedded in the historical and cultural context of the society in which we live, therefore humans are called upon to act ethically in order to decide well and not only in terms of efficiency.

The data generation process begins with data collection from the world. The process involves both sampling from a population and identifying which features and labels to use (representation and measurement bias). Data is collected into benchamrk data used to evaluate, compare and motivate the development of better models (evaluation and aggregation bias). The final model then generates its output, which has some real wordld implications because it it integrated into system through human interpretation (deployment bias).

- How can human-designed machines be expected not to be influenced by inherent social prejudices?

[#10] Any ideas on different ways to reduce the bias in this specific problems?

The last task will focus on reflection about the problem and possible developments underling difficulties and the tradeoffs encountered.

Algorithmic decision-making can reduce face-to-face discrimination in markets prone to implicit and explicit biases. But the use of algorithms can also lead to inadvertent discrimination (Barocas and Selbst, 2016).

The explosion of data, coupled with the significant growth of ML and AI, offers a huge opportunity to correct substantial problems in the current system of mortgage allocation thanks to the adoption of automatic decision based system for the evaluation of solvency. Lenders have used the legitimate-business-necessity defense to argue that any variable that is correlated with default is acceptable. Anti-discrimination laws currently in effect are inapplicable to this situation. But by just opening the floodgates in accordance with the principle that "you can do better than today," a Pandora's box of brand-new issues is unlocked. Artificial intelligence (AI) offers the opportunity to transform the means we allocate credit and risk and to create fairer and more inclusive systems. However, artificial intelligence can easily go in the opposite direction, exacerbating existing biases, creating cycles that reinforce the skewed allocation of credit and making discrimination in lending even harder to find. If a model is trained on an unfair dataset, such as one in which a higher percentage of married borrowers defaulted on their loans than single borrowers with the same income, annuity loan, etc., such biases will affect the model's predictions when applied to real-world situations. Lending discrimination occurs when lenders base credit decisions on factors other than the applicant’s creditworthiness. This is one factor behind the racial wealth gap that persists in the United States today. This discriminatory practice for example, made it impossible for many members of racial and ethnic minority groups to qualify for mortgage loans. In the United States laws today forbid discrimination based on race, color, religion, sex, national origin, handicap, familial status, age, and whether you receive public assistance income.

If past decisions were biased also the data transcripted, then automated systems that learn from historical data will also be biased.Training on historical data would perpetuate that injustice. This is crucial, is a paradox, a vicious circle. Automated decisions can be faster, cheaper, and less subjective than manual ones but bias in models can have far-reaching societal consequences, like worsening wealth inequality.

Gender is a social construct (World Health Organization). There is no basis for classifying people according to gender, but gender prejudice and discrimination have very real effects. A similar argument can be made for all ethical variables that do not directly influence customers' propensity to pay. Every year, financial institutions are required to publicly disclose loan-level information about their mortgages through the Home Mortgage Disclosure Act (HMDA). The latest HMDA data makes it clear: women pay more than men for their mortgages in every state in the U.S. except Alaska (Home Mortgage Disclosure Act, 2021). The inequity in rates is indicative of a systemic problem and deeper underlying factors. One reason may be that, in the absence of pay equity, women may find themselves with lower debt-to-income ratios, lower credit scores and, consequently, higher mortgage rates than men, which may translate into a more likely difficulty in meeting payment deadlines with the bank.

As a first step, we propose that the ethical variables we described above be divided into 2 subgroups Ethical variables that are not relevant in assessing creditworthiness as they do not affect payment difficulties such as: Gender, Age of the client, Companion, Number of the region's population, Days employed. From now on we will call these, strong ethical variables.
The remaining ethical variables that might be relevant in assessing solvency we will call weak ethical variables

We believe that strong ethical variables should not really be collected by data controllers and processed. Taking the variable 'Gender' as an example, we see no case where this category can or should influence the classification. An anologous discussion can be made for the other strong variables. As far as weak ethical variables are concerned, we believe that when talking about money and propensity to meet payment deadlines, the bank should firstly base its judgement on variables affecting the customer's propensity to pay and secondly on weak ethical variables. Weak ethical variables such as Occupation type or Education type are objectively a likely indicator of the customer's propensity to pay but could lead to a discriminatory bias.

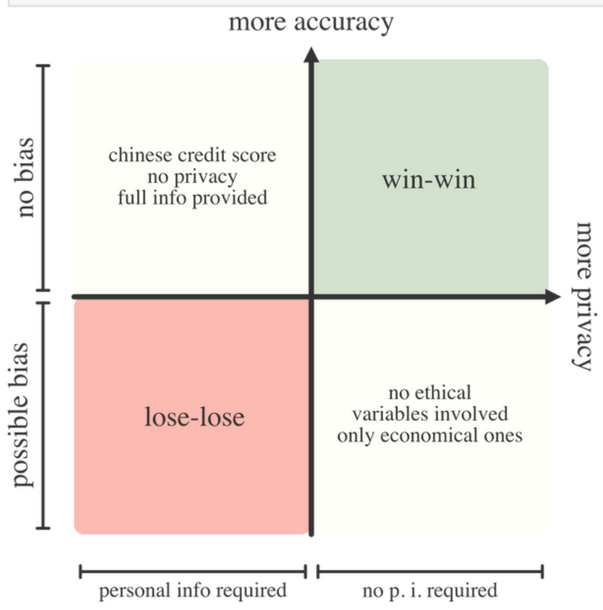
Returning to the algorithm, it is difficult to judge positively an algorithm that has such an important and delicate objective but takes into account so few variables (25), most of which we consider prone to ethical discussion. Furthermore, considering the results of the feature selections, we see how variables that should be absolutely irrelevant to the judgement, such as the gender of the individual, acquire such considerable importance as to make it the third most important feature in the model.

Framework:

We created this framework where we examine the trade-off between accuracy (expressed on the y-axis) and privacy where we mean the level of personal information required by the bank with the aim of eliminating bias in the model (represented on the x-axis). The first important realization is that the current system is located at the point where the axes on which we are trading cross, or the graph's origin. This compels decision-makers to question whether implementing a new system that has some distortions but less than those of the current system would constitute progress. Although it could be challenging to accept a structure that is fundamentally warped, it is crucial to understand that the current situation is already very distorted. Therefore, rejecting a new technology because it contains a certain level of bias does not mean we are protecting the system from bias. On the contrary, it may mean that we are allowing a more distorted system to perpetuate itself.

In [73]:
```
Image("framework.png", width=300, height=300)
```

Out[73]:



Lose Lose

As shown in the figure above, the bottom left corner (quadrant III) is one where AI results in a system that request more information and less predictive accuracy. Regulation and commercial incentives should work together against this outcome. It may be difficult to imagine incorporating more information that reduces accuracy, but it is not inconceivable. The data used in the real world are not as pure as those model testing. Potential occurrence of policy moving in this direction is the introduction of inaccurate or misleading data that may confuse an AI into thinking it has increased accuracy when it has not.

Trade-offs: More accuracy but less fairness

AI that increases accuracy but do not respect privacy gets a lot of attention, deservedly so. This scenario represented in the top left (quadrant II) of this framework can range from the introduction of data that are clear proxies for protected classes to information.

The new AI may increase access to credit on better terms than under the current system, but it is still not permissible to allow a system to access to all the personal information and to be accepted simply because it is less biased than previous discriminatory practices.

Sesame Credit Case (Dr. Fei Shen, GovInsederAsia):

New social credit products introduced by large Internet companies have emerged in recent years. The best known case is Sesame Credit (Zhima Credit). Sesame Credit is a private credit scoring system developed by Ant Financial Services Group (a subsidiary of Alibaba Group). The Sesame Credit programme was launched in 2015, but Sesame Credit's data comes mainly from Alibaba's Alipay, which was launched in 2003. The data generated on the Alipay platform includes loans, payments, purchases and insurance records.

Technically speaking, Sesame Credit is a functional component embedded in Alipay, a third-party online payment platform. Currently, there are about 520 million users of the service. Sesame Credit provides a score for individual users. The score is derived from five dimensions: credit history, ability to fulfil, personal characteristics, behaviour and preferences, and interpersonal relationships.

No specific explanation is given by Alipay on how a concrete score is calculated from the records from the five dimensions. It seems that data on credit history, ability to evade and behaviour and preferences come from one's own transaction data on Alipay. Data on personal characteristics are optional and supplemented by the users themselves. They include education level, driving licence and vehicle registration information, etc. The last category, interpersonal relationships, seems rather scary and strange. It implies that if you have friends with a good credit score, then you will also be a good individual. Conversely, if your social network is full of unreliable friends, your score will be lower. However, algorithms are not transparent.

The implementation of this dystopian scenario, which follows in the footsteps of science fiction works such as 1984 by George Orwell, would mean having to give the bank as much personal information as it needs to judge our creditworthiness. How many of us would be willing to let the bank know information from our private sphere if this would result in an extremely accurate judgement by the machine? The machine could uncover correlations between insolvency and daily habits that we were unaware of and that could influence our lives. This credit system extends this idea to all aspects of life, judging the behaviour and reliability of the policyholder.

This judging credit system will hypothetically bring benefits to the financial system due to the extreme precision that the algorithm will achieve due to the breadth of variables to be taken into account, but it is important to highlight the dystopian-sounding nature of the bank that assumes a role in monitoring and the behaviour of citizens.

Trade-offs: Less accuracy but more fairness

The bottom right (quadrant IV) represents the trade-offs that leads to an increase in level of privacy but a decrease in accuracy

Let us assume that only strictly economic variables are used in the algorithm, completely excluding any ethical variables. Specifically, the algorithm will predict on customer information such as income, house value, assets, debts/debts to financial institutions, salary etc..

Due to the correlation between wealth and income and the racial, gender, and other protected classifications, cash flow analysis is somewhat biased. However, the current fair-lending system should have no trouble allowing for a more intelligent use of this information since income and wealth are acceptable existing factors in the evaluation of solvency.

Industry pressures would often push back against such limitations and support higher accuracy. To avoid bias, social principles of fairness may call for compromises on accuracy. However, limiting how this information is used will not solve the issue. On the other hand, society's attempts to restrict the use of data for purposes of justice may have difficulties as a result of AI's capacity to uncover hidden proxies for that data. Problems that seem to be solved by bans then simply migrate into the world of algorithms, where they reappear.

A win-win situation Quadrant I, top right, represents the incorporation of artificial intelligence that increases accuracy while maintaining a high level of privacy for the customer. At first glance, this should be a win-win situation. The industry assigns credit more accurately, increasing respect for customer information. Consumers enjoy greater availability of credit on more accurate terms and with less bias than the existing status quo. This optimistic scenario is assumed to be achievable by retraining the data collected and supplied to the algorithm.

Our method, explained below, uses pre-processing to find and eliminate biased datapoints from the training data while maintaining the integrity of all other datapoints. We hypothesize that certain training data is more biased than the rest and, as a result, has a greater impact on a learnt model's predictions.

After normalisation of all economic variables: Find instances where a learnt model has made a mistake. In particular, create a pool of comparable individual pairs first, then identify the discriminating pairs within it. Then, heuristically ascertain which of the two people in each discriminating pair was not treated fairly. Sort the training data points in order of how much each one contributed to the unfair judgments. Retrain a model with a lower individual discrimination by removing some of the top-ranked training datapoints.

In the example below, where 1 represents the granting of the loan and 0 the non-granting, we note a typical situation of observation bias as even having more favourable economic values than others, customer number 2 is denied the loan due to the ethical variable gender.

In [74]:
```
Image("Table.png", width=300, height=300)
```

Out[74]:

	INCOME	GOODS_PRICE	GENDER	DECISION
#1	1	0,1	M	1
#2	0,9	0,7	F	0
#3	0,8	0,3	M	1
#4	0,1	0,7	F	0
#5	0,1	0,5	M	0
#6	0,5	0,9	F	0
#7	1	0,8	F	1

We propose that the most biased datapoints are those that contribute most to the different forecasts. A debiased dataset is produced by removing the most significant data points (like the #2 observation in the example). A model developed using the debiased dataset is more equitable and discriminates against individuals less. Although it is preferable for loan approvals to be independent of the applicant's gender, previous loan approvers may have harbored unconscious or conscious biases.

The bank can train their model by missing the sensitive characteristic (for example, gender) to avoid differential treatment after discovering and eliminating the biased decisions. Our strategy gives the bank the ability to make future decisions that are less biased. By doing this, anti-discrimination rules are not broken, and it is also the correct thing to do.

Conclusion

Building ethical machine learning models is necessary for compliance with anti-discrimination regulations, as well as the morally just thing to do as it produces more desirable results like increased profitability. Injustice would be sustained by teaching a machine learning model biased historical judgments. This method enhances test accuracy, individual discrimination, and statistical disparity compared to a baseline model that is trained on historical data without eliminating any datapoints.