Анализ данных

1. выполнить анализ и охарактеризовать клиентский портфель организации
2. построить базовую модель прогнозирования банкротства, одобряющую не менее 35% клиентов при банкротстве среди одобренных не выше 15%.
3. подготовить рекомендации и предложения по изменению признакового пространства, использованию внешних данных и иному развитию базовой модели.

Описание признакового пространства

- **age** - Возраст заемщика
- **lastcredit** - Время в днях, которое прошло с момента открытия последнего кредитного продукта
- **time_to_lastcredit_closeddt** - Время в днях, которое прошло с момента закрытия последнего микрокредита (если есть активные кредиты, эта переменная будет равна 0)
- **close_loan_median** - Медиана, взятая по количеству дней между закрытием предыдущего и следующего микрокредита (считается по всем последовательно открытым микрозаймам), т.е. сколько в среднем проходит времени между закрытием предыдущего и следующего микрокредита
- **open_loan_median** - Медиана, взятая по количеству дней между открытием предыдущего и следующего микрокредита (считается по всем последовательно открытым микрозаймам), т.е. сколько в среднем проходит времени между открытием предыдущего и следующего микрокредита
- **is_active_100** - Количество активных кредитов, открытых за все время
- **isnt_active_100** - Доля не возвращенных кредитов относительно всех кредитов, взятых за все время
- **is_lost_100** - Невозвращенные кредиты, открытые за последний месяц (например, переданные по цессии)
- **micro_loans_active_100** - Активная сумма микрокредитов, открытых за всё время
- **is_active_12** - Количество всех активных кредитов, открытых за последние 12 месяцев
- **open_sum_12** - Активная сумма кредитов, взятых за последние 12 месяцев
- **isnt_active_12** - Количество закрытых кредитов, которые были открыты за последние 12 месяцев
- **is_lost_12** - Невозвращенные кредиты, открытые за последние 12 месяцев
- **overdue_loans_12** - Количество просроченных кредитов, открытых за последние 12 месяцев
- **micro_loans_active_12** - Активная сумма микрокредитов, открытых за последние 12 месяцев
- **is_active_3** - Количество всех активных кредитов, открытых за последние 3 месяца
- **open_sum_3** - Активная сумма кредитов, взятых за последние 3 месяца
- **isnt_active_3** - Количество закрытых кредитов, которые были открыты за последние 3 месяца
- **is_lost_3** - Невозвращенные кредиты, открытые за последние 3 месяца
- **overdue_loans_3** - Количество просроченных кредитов, открытых за последние 3 месяца
- **micro_loans_active_3** - Активная сумма микрокредитов, открытых за последние 3 месяца
- **is_active_1** - Количество всех активных кредитов, открытых за последний месяц
- **open_sum_1** - Активная сумма кредитов, взятых за последний месяц
- **isnt_active_1** - Количество закрытых кредитов, которые были открыты за последний месяц
- **is_lost_1** - Невозвращенные кредиты, открытые за последний месяц (например, переданные по цессии)
- **micro_loans_active_1** - Активная сумма микрокредитов, открытых за последний месяц
- **ratio_all_microloans_3_to_12** - Отношение количества микрокредитов, взятых за последние 3 месяца, к количеству микрокредитов, взятых за последние 12 месяцев

- **ratio_overdue_loans_3_to_12** - Отношение количества просроченных микрокредитов, взятых за последние 3 месяца, к количеству просроченных микрокредитов, взятых за последние 12 месяцев
- **ratio_history_100** - Доля не возвращенных кредитов относительно всех кредитов, взятых за все время
- **ratio_history_12** - Доля не возвращенных кредитов относительно всех кредитов, взятых за последние 12 месяцев
- **fraction_last_x_12** - Доля кредитов, взятых за последние 12 месяцев, относительно всех кредитов истории
- **ratio_history_3** - Доля не возвращенных кредитов относительно всех кредитов, взятых за последние 3 месяца
- **fraction_last_x_3** - Доля кредитов, взятых за последние 3 месяца, относительно всех кредитов истории
- **ratio_history_1** - Доля не возвращенных кредитов относительно всех кредитов, взятых за последний месяц
- **fraction_last_x_1** - Доля кредитов, взятых за последний месяц, относительно всех кредитов истории
- **mean_delay_100_with_lag** - Средняя просрочка за всё время (с лагом по времени в 2 месяца)
- **mean_delay_12_with_lag** - Средняя просрочка за последние 12 месяцев (с лагом по времени в 2 месяца)
- **mean_delay_3_with_lag** - Средняя просрочка за последние 3 месяца (с лагом по времени в 2 месяца)
- **mean_delay_1_with_lag** - Средняя просрочка за последний месяц (с лагом по времени в 2 месяца)
- **ratio_mean_delay_3_to_12** - Отношение средней просрочки за последние 3 месяца к средней просрочке за последние 12 месяцев (в днях, с лагом по времени в 2 месяца)
- **count_all_credits** - Количество всех кредитов в истории
- **ratio_pattern_len_to_pattern_1** - Отношение количества платежей в платежном паттерне к общему количеству запланированных платежей на данный момент
- **ratio_pattern_len_to_pattern_2** - Отношение количества просрочек в 0-5 дней в платежном паттерне к общему количеству запланированных платежей на данный момент
- **ratio_pattern_len_to_pattern_3** -
- **ratio_pattern_len_to_pattern_4** -
- **ratio_pattern_len_to_pattern_bad_len** - Отношение количества символов сильной просрочки (> 60 дней) в платежном паттерне к общему количеству символов в строке
- **last_microloan_openeddt** - Время в днях, которое прошло с момента открытия последнего микрокредита
- **is_type_credit_card_100** - Количество кредитов типа "кредитная карта", открытых за всё время
- **is_type_consumer_100** - Количество кредитов типа "потребительский кредит", открытых за всё время
- **is_type_micro_100** - Количество кредитов типа "микрокредит", открытых за всё время
- **is_active_type_credit_card_100** - Количество активных кредитов, открытых за всё время с типом займа "кредитная карта"
- **is_active_type_consumer_100** - Количество активных кредитов, открытых за всё время с типом займа "потребительский кредит"
- **is_active_type_micro_100** - Количество активных кредитов, открытых за все время с типом займа "микрокредит"
- **is_type_credit_card_12** - Количество кредитов типа "кредитная карта", открытых за последние 12 месяцев
- **is_type_consumer_12** - Количество кредитов типа "потребительский кредит", открытых за последние 12 месяцев
- **is_type_micro_12** - Количество кредитов типа "микрокредит", открытых за последние 12 месяцев

- **is_active_type_credit_card_12** - Количество активных кредитов, открытых за последние 12 месяцев с типом займа "кредитная карта"
- **is_active_type_consumer_12** - Количество активных кредитов, открытых за последние 12 месяцев с типом займа "потребительский кредит"
- **is_active_type_micro_12** - Количество активных кредитов, открытых за последние 12 месяцев с типом займа "микрокредит"
- **is_type_credit_card_3** - Количество кредитов типа "кредитная карта", открытых за последние 3 месяца
- **is_type_consumer_3** - Количество кредитов типа "потребительский кредит", открытых за последние 3 месяца
- **is_type_micro_3** - Количество кредитов типа "микрокредит", открытых за последние 3 месяца
- **is_active_type_credit_card_3** - Количество активных кредитов, открытых за последние 3 месяца с типом займа "кредитная карта"
- **is_active_type_consumer_3** - Количество активных кредитов, открытых за последние 3 месяца с типом займа "потребительский кредит"
- **is_active_type_micro_3** - Количество активных кредитов, открытых за последние 3 месяца с типом займа "микрокредит"
- **is_type_credit_card_1** - Количество кредитов типа "кредитная карта", открытых за последние 12 месяцев
- **is_type_consumer_1** - Количество кредитов типа "потребительский кредит", открытых за последний месяц
- **is_type_micro_1** - Количество кредитов типа "микрокредит", открытых за последний месяц
- **is_active_type_credit_card_1** - Количество активных кредитов, открытых за последние 12 месяцев с типом займа "кредитная карта"
- **is_active_type_consumer_1** - Количество активных кредитов, открытых за последний 1 месяц с типом займа "потребительский кредит"
- **is_active_type_micro_1** - Количество активных кредитов, открытых за последний 1 месяц с типом займа "микрокредит"
- **overall_worst_overdue_state_12** -
- **ratio_sum_outstanding_to_open_sum** - Отношение суммы просрочки по всем кредитам к сумме взятых кредитов за всю историю

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import shap

from sklearn.model_selection import KFold, cross_val_score, train_test_split
from sklearn.metrics import r2_score, f1_score, confusion_matrix, classification_report
from sklearn.ensemble import RandomForestRegressor
import xgboost as xgb


pd.set_option('display.max_columns', None)
pd.set_option('display.float_format', lambda x: '%.5f' % x)
%matplotlib inline
```

In [2]:

```
# ф-я пишет сколько пропусков и какой процент от общего числа строк пропущенно
def chech_omissions(data):
    print("всего строк: {}".format(data.shape[0]))
    for i, col in list(zip(data.isna().sum(),data.columns)):
        print("{}: кол-во пропусков: {}, процент: {} % ".format(col,i , round(i/data.sha
```

In [3]:

```
# корреляция
def cor(data, target):
    correlation = data.corr()
    corr_with_target = correlation[target].sort_values(ascending = False)
    print(corr_with_target)
    fig, axes = plt.subplots(figsize = (14,12))
    plt.title("Correlation of Numeric Features with target", y=1, size=16)
    sns.heatmap(correlation, square=True, vmax=0.8, cmap="viridis")
    return corr_with_target, correlation
```

In [4]:

```
# ф-я определяет линейную зависимость между признаками, порог 0.5
def check_signs_corr(table_corr):
    corr_row_col = {}
    for i in table_corr.iterrows():
    #     print(i[1])
        dict_ = dict(i[1])
    #     print(dict_)
        for k, v in dict_.items():
            if v > 0.5 and i[0] != k:
                corr_row_col[v] = [i[0], k]

    return corr_row_col
```

In [5]:

```
# ф-я удаляет столбцы коррелирующие между собой
def del_columns_corr(data_df, dict_signs_corr):
    list_columns = []
    data = data_df.copy()
    for val in dict_signs_corr.values():
        for col in val:
            if col in data.columns and col not in list_columns:
                ind_col = val.index(col)
                if ind_col == 0:
                    list_columns.append(val[1])
                    data.drop(col, inplace=True, axis=1)
                elif ind_col == 1:
                    list_columns.append(val[0])
                    data.drop(col, inplace=True, axis=1)

    return data
```

In [6]:

```
# ф-я заменяет пропуски с нужным порогом на медиану
def fillna_omissions(data_df, p= 0):
    data = data_df.copy()
#     print("Всего строк: {}".format(data.shape[0]))
    for i, col in list(zip(data.isna().sum(),data.columns)):
        if i> p:
            print("{}: кол-во пропусков замененных на медиану: {} ".format(col,i,1))
#             print(percent, col)
            data[col].fillna(data[col].median(), inplace=True)
    return data
```

In [7]:

```
"""
ф-я заменяет бесконечность на медиану
"""
def fillna_inf(data_df):
    data = data_df.copy()
    for i in data.columns:
        inf_col = np.where(abs(data[i]) == np.inf)

        if len(list(inf_col)[0]) > 0:
            data.loc[data[i] == np.inf, i] = data[i].median()
            data.loc[data[i] == -np.inf, i] = data[i].median()
    return data
```

In [8]:

```
"""
ф-я определяет нелинейную связь признаков, возвращает список из 15 наиболее важных
"""
def nonlianer_comun(data, target="target")-> list:
    parameters = {"max_depth": 6, "n_estimators": 25, "random_state": 27, "n_jobs": 2}

    forest = RandomForestRegressor(**parameters)
    forest.fit(data.drop(target, axis=1), data[target])

    # numerical original
    top = []
    n_top = 15
    importances = forest.feature_importances_
    idx = np.argsort(importances)[::-1][0:n_top]
    feature_names = data.drop(target, axis=1).columns

    plt.figure(figsize=(20, 5))
    sns.barplot(x=feature_names[idx], y=importances[idx], palette="viridis")
    top.append(feature_names[idx])
    plt.title("What are the top important features to start with?", size=14)
    plt.xticks(rotation=90)
    return top
```

In [9]:

```
def model_fit(model_f, data, param=False, random_state=42):
    x_train, x_test, y_train, y_test = train_test_split(data.drop("target", axis=1), dat
    print("x_train.shape: ", x_train.shape)
    print("y_train.shape: ", y_train.shape)
    print("x_test.shape: ", x_test.shape)
    print("y_test.shape: ", y_test.shape)
    if param:
        model = model_f(params= param, random_state=random_state)
    else:
        model = model_f(random_state=random_state)
    model.fit(x_train, y_train)
    pred = model.predict(x_test)
    test_score = confusion_matrix(y_test, pred)

    ax= plt.subplot()
    sns.heatmap(test_score, annot=True, fmt='g', ax=ax);  #annot=True to annotate cells,

    # labels, title and ticks
    ax.set_xlabel('Predicted labels');ax.set_ylabel('True labels');
    ax.set_title('Confusion Matrix');
    ax.xaxis.set_ticklabels(['True', 'False']); ax.yaxis.set_ticklabels(['True', 'False'

    print(classification_report(y_test, pred))
    return model, x_train, x_test
```

Базовый анализ данных

In [10]:

```
cl = pd.read_csv("data/all_reject_data.csv")
cr = pd.read_csv("data/new_training_data_31_08_2022.csv")

print("cl rows: {}, columns: {}".format(*cl.shape))
print("cr rows: {}, columns: {}".format(*cr.shape))
```

```
cl rows: 106511, columns: 74
cr rows: 42529, columns: 75
```

In [11]:

```
cl.head(n=3)
```

Out[11]:

	Unnamed: 0	age	lastcredit	time_to_lastcredit_closeddt	close_loan_median	open_loa
0	4	35.00000	903.00000	44683.00000	0.00000	43
1	6	32.00000	4.00000	0.00000	0.00000	
2	9	25.00000	17.00000	0.00000	0.00000	

In [12]:

```
cr.head(n=3)
```

Out[12]:

	Unnamed: 0	age	lastcredit	time_to_lastcredit_closeddt	close_loan_median	open_loar
0	0	43.00000	81.00000	235.00000	0.00000	
1	1	35.00000	3.00000	0.00000	0.00000	
2	2	27.00000	19.00000	65.00000	0.00000	

In [13]:

```
cl.rename(columns={"Unnamed: 0": "id"}, inplace=True)
cr.rename(columns={"Unnamed: 0": "id"}, inplace=True)
```

In [14]:

```
cr.describe()
```

Out[14]:

	id	age	lastcredit	time_to_lastcredit_closeddt	close_loan_media
count	42529.00000	42529.00000	42529.00000	42529.00000	42520.0000
mean	16848.10764	34.81008	10.63573	127.21907	0.3418
std	10120.47745	8.82418	75.48064	2343.82402	13.0845
min	0.00000	18.00000	1.00000	0.00000	0.0000
25%	8134.00000	28.00000	2.00000	0.00000	0.0000
50%	16522.00000	33.00000	4.00000	0.00000	0.0000
75%	25052.00000	39.00000	7.00000	0.00000	0.0000
max	35684.00000	75.00000	5029.00000	44785.00000	1428.5000

По таблице сверху видно, что организация чаще выдает микрозаймы, чем карточки или потребительские кредиты

In [15]:

```
count_days = cr.time_to_lastcredit_closeddt.value_counts()
count_days
```

Out[15]:

```
0.00000       40674
2.00000         103
3.00000          83
4.00000          82
1.00000          72
              ...
187.00000         1
200.00000         1
190.00000         1
449.00000         1
2094.00000        1
Name: time_to_lastcredit_closeddt, Length: 401, dtype: int64
```

В основном все кредиты в кредитной организации закрыты

посмотрим на возраст людей наиболее часто берущих займы

In [16]:

```
cr["age"].value_counts()
```

Out[16]:

```
32.00000    2235
33.00000    2162
31.00000    2152
35.00000    2089
30.00000    2046
29.00000    2009
34.00000    2002
28.00000    1839
36.00000    1827
27.00000    1788
37.00000    1737
26.00000    1644
38.00000    1587
39.00000    1455
25.00000    1378
40.00000    1295
24.00000    1270
23.00000    1128
41.00000    1117
42.00000     957
22.00000     882
43.00000     876
44.00000     787
45.00000     558
46.00000     538
21.00000     510
47.00000     503
48.00000     486
49.00000     440
51.00000     372
50.00000     343
52.00000     297
53.00000     286
55.00000     216
54.00000     197
57.00000     163
20.00000     161
56.00000     139
59.00000     135
58.00000     124
60.00000     123
61.00000     114
19.00000      84
62.00000      80
63.00000      73
64.00000      71
65.00000      56
66.00000      37
68.00000      34
69.00000      32
67.00000      27
70.00000      20
18.00000      19
71.00000      13
73.00000      10
72.00000       4
75.00000       1
74.00000       1
Name: age, dtype: int64
```

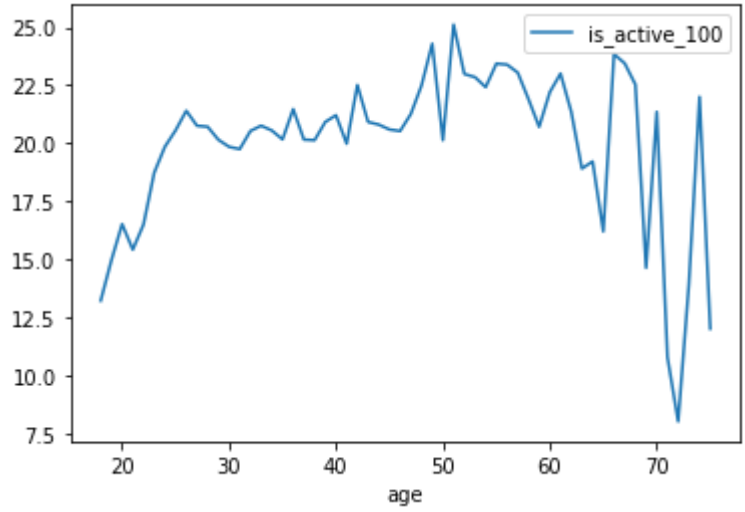
Видно, что самое большое кол-во человек когда либо берущих займы в кредитной организации это люди в возрасте от 25 до 40 лет, тк их больше всего

In [17]:

```
cr_age = cr[["age","is_active_100"]].groupby(["age"]).mean()
cr_age.plot()
```

Out[17]:

```
<AxesSubplot:xlabel='age'>
```



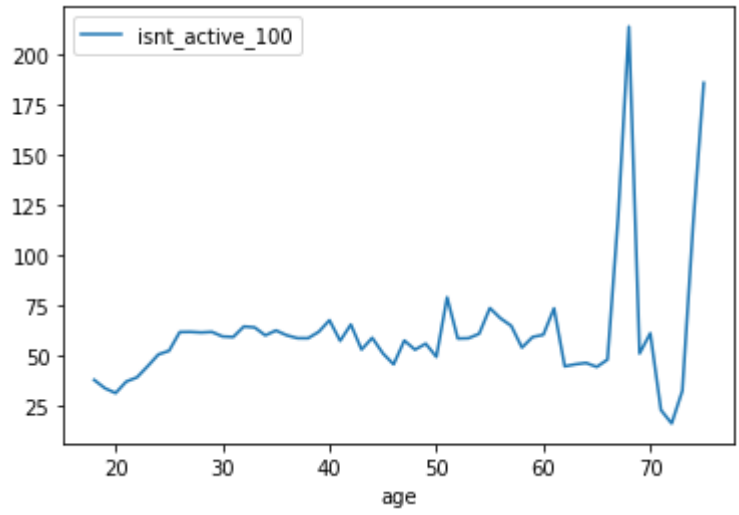
Активных кредитов в основном у людей в возрасте от 25 до 60

In [18]:

```
cr_age = cr[["age","isnt_active_100"]].groupby(["age"]).mean()
cr_age.plot()
```

Out[18]:

```
<AxesSubplot:xlabel='age'>
```



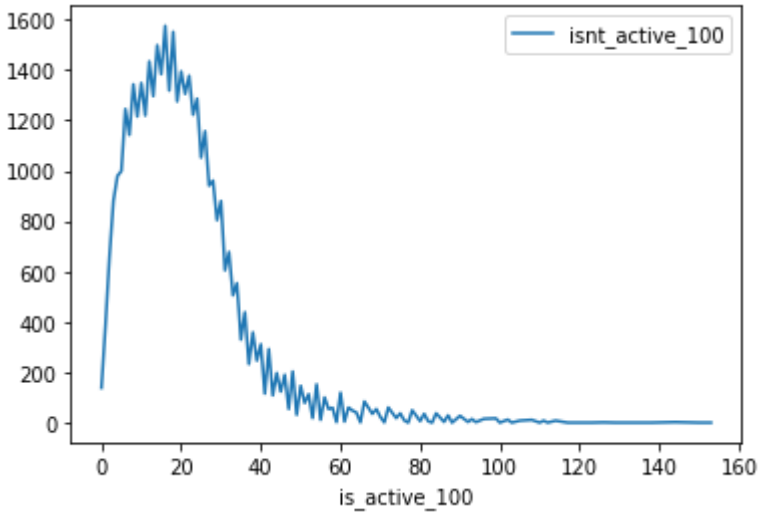
Интересно, что люди преклоннго возраста реже возвращают деньги

In [19]:

```
cr_active = cr[["is_active_100","isnt_active_100"]].groupby(["is_active_100"]).count()
cr_active.plot()
```

Out[19]:

```
<AxesSubplot:xlabel='is_active_100'>
```



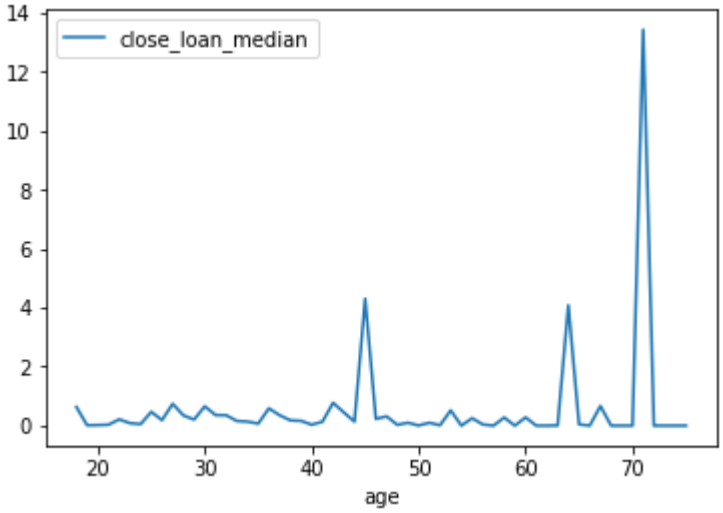
По данному графику можно предположить, что люди стараются быстрее закрыть кредит

In [20]:

```
cr_age_close = cr[["age","close_loan_median"]].groupby(["age"]).mean()
cr_age_close.plot()
```

Out[20]:

```
<AxesSubplot:xlabel='age'>
```



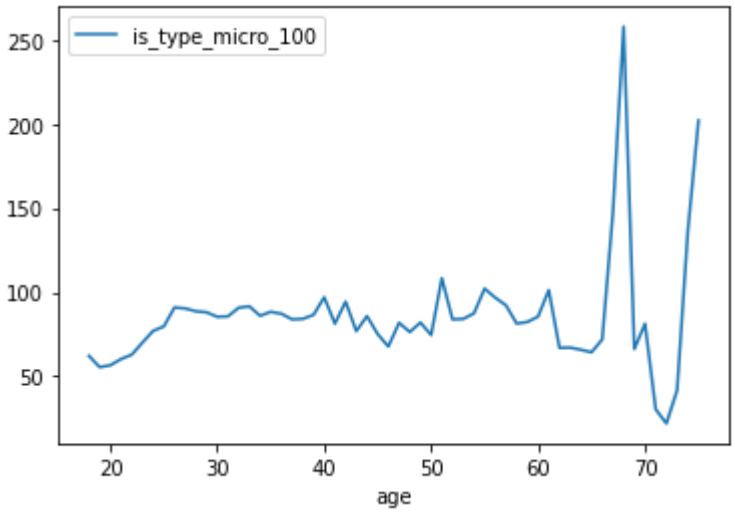
Люди 40-50, 60- , наиболее большой перерыв между закрытым и заного взятым кредитом

In [21]:

```
cr_age = cr[["age", "is_type_micro_100"]].groupby(["age"]).mean()
cr_age.plot()
```

Out[21]:

```
<AxesSubplot:xlabel='age'>
```



In [22]:

```
cr["is_type_micro_100"].value_counts()
```

Out[22]:

```
30.00000     487
34.00000     458
32.00000     456
44.00000     441
28.00000     438
            ...
576.00000      1
781.00000      1
656.00000      1
415.00000      1
674.00000      1
Name: is_type_micro_100, Length: 748, dtype: int64
```

In [23]:

```
cr[["is_type_micro_100", "age"]][(cr['is_type_micro_100'] >= 30.0) &
                                 (cr['is_type_micro_100'] <= 40.0)].sort_values(by="age'
```

Out[23]:

	is_type_micro_100	age
4733	33.00000	18.00000
5987	35.00000	18.00000
5986	35.00000	18.00000
580	35.00000	18.00000
1537	33.00000	19.00000
390	36.00000	19.00000
3295	35.00000	19.00000
2245	34.00000	19.00000
1579	34.00000	19.00000
315	38.00000	19.00000
6653	34.00000	19.00000
2671	32.00000	19.00000
3946	31.00000	19.00000
1338	34.00000	19.00000
1630	39.00000	19.00000
2847	40.00000	19.00000
1955	37.00000	19.00000
2742	40.00000	19.00000
4764	39.00000	19.00000
887	34.00000	19.00000

```
от 30 тр до 40 тр , в такой сумме чаще нуждаются люди
```

In [27]:

```
# Посмотрим пропуски
chech_omissions(cl)
```

```
всего строк: 106511
id: кол-во пропусков: 0, процент: 0 %
age: кол-во пропусков: 0, процент: 0 %
lastcredit: кол-во пропусков: 0, процент: 0 %
time_to_lastcredit_closeddt: кол-во пропусков: 0, процент: 0 %
close_loan_median: кол-во пропусков: 138, процент: 0 %
open_loan_median: кол-во пропусков: 138, процент: 0 %
is_active_100: кол-во пропусков: 0, процент: 0 %
isnt_active_100: кол-во пропусков: 0, процент: 0 %
is_lost_100: кол-во пропусков: 0, процент: 0 %
micro_loans_active_100: кол-во пропусков: 0, процент: 0 %
is_active_12: кол-во пропусков: 0, процент: 0 %
open_sum_12: кол-во пропусков: 0, процент: 0 %
isnt_active_12: кол-во пропусков: 0, процент: 0 %
is_lost_12: кол-во пропусков: 0, процент: 0 %
overdue_loans_12: кол-во пропусков: 0, процент: 0 %
micro_loans_active_12: кол-во пропусков: 0, процент: 0 %
is_active_3: кол-во пропусков: 0, процент: 0 %
open_sum_3: кол-во пропусков: 0, процент: 0 %
isnt_active_3: кол-во пропусков: 0, процент: 0 %
is_lost_3: кол-во пропусков: 0, процент: 0 %
overdue_loans_3: кол-во пропусков: 0, процент: 0 %
micro_loans_active_3: кол-во пропусков: 0, процент: 0 %
is_active_1: кол-во пропусков: 0, процент: 0 %
open_sum_1: кол-во пропусков: 0, процент: 0 %
isnt_active_1: кол-во пропусков: 0, процент: 0 %
is_lost_1: кол-во пропусков: 0, процент: 0 %
micro_loans_active_1: кол-во пропусков: 0, процент: 0 %
ratio_all_microloans_3_to_12: кол-во пропусков: 589, процент: 1 %
ratio_overdue_loans_3_to_12: кол-во пропусков: 14863, процент: 14 %
ratio_history_100: кол-во пропусков: 27, процент: 0 %
ratio_history_12: кол-во пропусков: 491, процент: 0 %
fraction_last_x_12: кол-во пропусков: 27, процент: 0 %
ratio_history_3: кол-во пропусков: 2384, процент: 2 %
fraction_last_x_3: кол-во пропусков: 27, процент: 0 %
ratio_history_1: кол-во пропусков: 3874, процент: 4 %
fraction_last_x_1: кол-во пропусков: 27, процент: 0 %
mean_delay_100_with_lag: кол-во пропусков: 0, процент: 0 %
mean_delay_12_with_lag: кол-во пропусков: 0, процент: 0 %
mean_delay_3_with_lag: кол-во пропусков: 0, процент: 0 %
mean_delay_1_with_lag: кол-во пропусков: 0, процент: 0 %
ratio_mean_delay_3_to_12: кол-во пропусков: 2903, процент: 3 %
count_all_credits: кол-во пропусков: 0, процент: 0 %
ratio_pattern_len_to_pattern_1: кол-во пропусков: 138, процент: 0 %
ratio_pattern_len_to_pattern_2: кол-во пропусков: 138, процент: 0 %
ratio_pattern_len_to_pattern_3: кол-во пропусков: 138, процент: 0 %
ratio_pattern_len_to_pattern_4: кол-во пропусков: 138, процент: 0 %
ratio_pattern_len_to_pattern_bad_len: кол-во пропусков: 138, процент: 0 %
last_microloan_openeddt: кол-во пропусков: 0, процент: 0 %
is_type_credit_card_100: кол-во пропусков: 0, процент: 0 %
is_type_consumer_100: кол-во пропусков: 0, процент: 0 %
is_type_micro_100: кол-во пропусков: 0, процент: 0 %
is_active_type_credit_card_100: кол-во пропусков: 0, процент: 0 %
is_active_type_consumer_100: кол-во пропусков: 0, процент: 0 %
is_active_type_micro_100: кол-во пропусков: 0, процент: 0 %
is_type_credit_card_12: кол-во пропусков: 0, процент: 0 %
is_type_consumer_12: кол-во пропусков: 0, процент: 0 %
is_type_micro_12: кол-во пропусков: 0, процент: 0 %
is_active_type_credit_card_12: кол-во пропусков: 0, процент: 0 %
is_active_type_consumer_12: кол-во пропусков: 0, процент: 0 %
is_active_type_micro_12: кол-во пропусков: 0, процент: 0 %
```

```
is_type_credit_card_3: кол-во пропусков: 0, процент: 0 %
is_type_consumer_3: кол-во пропусков: 0, процент: 0 %
is_type_micro_3: кол-во пропусков: 0, процент: 0 %
is_active_type_credit_card_3: кол-во пропусков: 0, процент: 0 %
is_active_type_consumer_3: кол-во пропусков: 0, процент: 0 %
is_active_type_micro_3: кол-во пропусков: 0, процент: 0 %
is_type_credit_card_1: кол-во пропусков: 0, процент: 0 %
is_type_consumer_1: кол-во пропусков: 0, процент: 0 %
is_type_micro_1: кол-во пропусков: 0, процент: 0 %
is_active_type_credit_card_1: кол-во пропусков: 0, процент: 0 %
is_active_type_consumer_1: кол-во пропусков: 0, процент: 0 %
is_active_type_micro_1: кол-во пропусков: 0, процент: 0 %
overall_worst_overdue_state_12: кол-во пропусков: 0, процент: 0 %
ratio_sum_outstanding_to_open_sum: кол-во пропусков: 1380, процент: 1 %
```

In [28]:

```
chech_omissions(cr)
```

```
всего строк: 42529
id: кол-во пропусков: 0, процент: 0 %
age: кол-во пропусков: 0, процент: 0 %
lastcredit: кол-во пропусков: 0, процент: 0 %
time_to_lastcredit_closeddt: кол-во пропусков: 0, процент: 0 %
close_loan_median: кол-во пропусков: 9, процент: 0 %
open_loan_median: кол-во пропусков: 9, процент: 0 %
is_active_100: кол-во пропусков: 0, процент: 0 %
isnt_active_100: кол-во пропусков: 0, процент: 0 %
is_lost_100: кол-во пропусков: 0, процент: 0 %
micro_loans_active_100: кол-во пропусков: 0, процент: 0 %
is_active_12: кол-во пропусков: 0, процент: 0 %
open_sum_12: кол-во пропусков: 0, процент: 0 %
isnt_active_12: кол-во пропусков: 0, процент: 0 %
is_lost_12: кол-во пропусков: 0, процент: 0 %
overdue_loans_12: кол-во пропусков: 0, процент: 0 %
micro_loans_active_12: кол-во пропусков: 0, процент: 0 %
is_active_3: кол-во пропусков: 0, процент: 0 %
open_sum_3: кол-во пропусков: 0, процент: 0 %
isnt_active_3: кол-во пропусков: 0, процент: 0 %
is_lost_3: кол-во пропусков: 0, процент: 0 %
overdue_loans_3: кол-во пропусков: 0, процент: 0 %
micro_loans_active_3: кол-во пропусков: 0, процент: 0 %
is_active_1: кол-во пропусков: 0, процент: 0 %
open_sum_1: кол-во пропусков: 0, процент: 0 %
isnt_active_1: кол-во пропусков: 0, процент: 0 %
is_lost_1: кол-во пропусков: 0, процент: 0 %
micro_loans_active_1: кол-во пропусков: 0, процент: 0 %
ratio_all_microloans_3_to_12: кол-во пропусков: 32, процент: 0 %
ratio_overdue_loans_3_to_12: кол-во пропусков: 6844, процент: 16 %
ratio_history_100: кол-во пропусков: 0, процент: 0 %
ratio_history_12: кол-во пропусков: 23, процент: 0 %
fraction_last_x_12: кол-во пропусков: 0, процент: 0 %
ratio_history_3: кол-во пропусков: 218, процент: 1 %
fraction_last_x_3: кол-во пропусков: 0, процент: 0 %
ratio_history_1: кол-во пропусков: 393, процент: 1 %
fraction_last_x_1: кол-во пропусков: 0, процент: 0 %
mean_delay_100_with_lag: кол-во пропусков: 0, процент: 0 %
mean_delay_12_with_lag: кол-во пропусков: 0, процент: 0 %
mean_delay_3_with_lag: кол-во пропусков: 0, процент: 0 %
mean_delay_1_with_lag: кол-во пропусков: 0, процент: 0 %
ratio_mean_delay_3_to_12: кол-во пропусков: 2561, процент: 6 %
count_all_credits: кол-во пропусков: 0, процент: 0 %
ratio_pattern_len_to_pattern_1: кол-во пропусков: 9, процент: 0 %
ratio_pattern_len_to_pattern_2: кол-во пропусков: 9, процент: 0 %
ratio_pattern_len_to_pattern_3: кол-во пропусков: 9, процент: 0 %
ratio_pattern_len_to_pattern_4: кол-во пропусков: 9, процент: 0 %
ratio_pattern_len_to_pattern_bad_len: кол-во пропусков: 9, процент: 0 %
last_microloan_openeddt: кол-во пропусков: 0, процент: 0 %
is_type_credit_card_100: кол-во пропусков: 0, процент: 0 %
is_type_consumer_100: кол-во пропусков: 0, процент: 0 %
is_type_micro_100: кол-во пропусков: 0, процент: 0 %
is_active_type_credit_card_100: кол-во пропусков: 0, процент: 0 %
is_active_type_consumer_100: кол-во пропусков: 0, процент: 0 %
is_active_type_micro_100: кол-во пропусков: 0, процент: 0 %
is_type_credit_card_12: кол-во пропусков: 0, процент: 0 %
is_type_consumer_12: кол-во пропусков: 0, процент: 0 %
is_type_micro_12: кол-во пропусков: 0, процент: 0 %
is_active_type_credit_card_12: кол-во пропусков: 0, процент: 0 %
is_active_type_consumer_12: кол-во пропусков: 0, процент: 0 %
is_active_type_micro_12: кол-во пропусков: 0, процент: 0 %
```

```
is_type_credit_card_3: кол-во пропусков: 0, процент: 0 %
is_type_consumer_3: кол-во пропусков: 0, процент: 0 %
is_type_micro_3: кол-во пропусков: 0, процент: 0 %
is_active_type_credit_card_3: кол-во пропусков: 0, процент: 0 %
is_active_type_consumer_3: кол-во пропусков: 0, процент: 0 %
is_active_type_micro_3: кол-во пропусков: 0, процент: 0 %
is_type_credit_card_1: кол-во пропусков: 0, процент: 0 %
is_type_consumer_1: кол-во пропусков: 0, процент: 0 %
is_type_micro_1: кол-во пропусков: 0, процент: 0 %
is_active_type_credit_card_1: кол-во пропусков: 0, процент: 0 %
is_active_type_consumer_1: кол-во пропусков: 0, процент: 0 %
is_active_type_micro_1: кол-во пропусков: 0, процент: 0 %
overall_worst_overdue_state_12: кол-во пропусков: 0, процент: 0 %
ratio_sum_outstanding_to_open_sum: кол-во пропусков: 2800, процент: 7 %
target: кол-во пропусков: 0, процент: 0 %
```

In [29]:

```
# меняю пропуски на медиану, т.к. пропусков мало
cr = fillna_omissions(cr)
```

```
close_loan_median: кол-во пропусков замененных на медиану: 9
open_loan_median: кол-во пропусков замененных на медиану: 9
ratio_all_microloans_3_to_12: кол-во пропусков замененных на медиану: 32
ratio_overdue_loans_3_to_12: кол-во пропусков замененных на медиану: 6844
ratio_history_12: кол-во пропусков замененных на медиану: 23
ratio_history_3: кол-во пропусков замененных на медиану: 218
ratio_history_1: кол-во пропусков замененных на медиану: 393
ratio_mean_delay_3_to_12: кол-во пропусков замененных на медиану: 2561
ratio_pattern_len_to_pattern_1: кол-во пропусков замененных на медиану: 9
ratio_pattern_len_to_pattern_2: кол-во пропусков замененных на медиану: 9
ratio_pattern_len_to_pattern_3: кол-во пропусков замененных на медиану: 9
ratio_pattern_len_to_pattern_4: кол-во пропусков замененных на медиану: 9
ratio_pattern_len_to_pattern_bad_len: кол-во пропусков замененных на меди
ану: 9
ratio_sum_outstanding_to_open_sum: кол-во пропусков замененных на медиан
у: 2800
```

In [30]:

```
# Проверяю есть-ли в данных значения бесконечности
for i in cr.columns:
        inf_col = np.where(abs(cr[i]) == np.inf) # беру модуль, тк может быть и минус бе
        print(i, inf_col, len(inf_col[0]))
```

```
id (array([], dtype=int64),) 0
age (array([], dtype=int64),) 0
lastcredit (array([], dtype=int64),) 0
time_to_lastcredit_closeddt (array([], dtype=int64),) 0
close_loan_median (array([], dtype=int64),) 0
open_loan_median (array([], dtype=int64),) 0
is_active_100 (array([], dtype=int64),) 0
isnt_active_100 (array([], dtype=int64),) 0
is_lost_100 (array([], dtype=int64),) 0
micro_loans_active_100 (array([], dtype=int64),) 0
is_active_12 (array([], dtype=int64),) 0
open_sum_12 (array([], dtype=int64),) 0
isnt_active_12 (array([], dtype=int64),) 0
is_lost_12 (array([], dtype=int64),) 0
overdue_loans_12 (array([], dtype=int64),) 0
micro_loans_active_12 (array([], dtype=int64),) 0
is_active_3 (array([], dtype=int64),) 0
open_sum_3 (array([], dtype=int64),) 0
isnt_active_3 (array([], dtype=int64),) 0
is_lost_3 (array([], dtype=int64),) 0
overdue_loans_3 (array([], dtype=int64),) 0
micro_loans_active_3 (array([], dtype=int64),) 0
is_active_1 (array([], dtype=int64),) 0
open_sum_1 (array([], dtype=int64),) 0
isnt_active_1 (array([], dtype=int64),) 0
is_lost_1 (array([], dtype=int64),) 0
micro_loans_active_1 (array([], dtype=int64),) 0
ratio_all_microloans_3_to_12 (array([], dtype=int64),) 0
ratio_overdue_loans_3_to_12 (array([], dtype=int64),) 0
ratio_history_100 (array([], dtype=int64),) 0
ratio_history_12 (array([], dtype=int64),) 0
fraction_last_x_12 (array([], dtype=int64),) 0
ratio_history_3 (array([], dtype=int64),) 0
fraction_last_x_3 (array([], dtype=int64),) 0
ratio_history_1 (array([], dtype=int64),) 0
fraction_last_x_1 (array([], dtype=int64),) 0
mean_delay_100_with_lag (array([], dtype=int64),) 0
mean_delay_12_with_lag (array([], dtype=int64),) 0
mean_delay_3_with_lag (array([], dtype=int64),) 0
mean_delay_1_with_lag (array([], dtype=int64),) 0
ratio_mean_delay_3_to_12 (array([], dtype=int64),) 0
count_all_credits (array([], dtype=int64),) 0
ratio_pattern_len_to_pattern_1 (array([], dtype=int64),) 0
ratio_pattern_len_to_pattern_2 (array([], dtype=int64),) 0
ratio_pattern_len_to_pattern_3 (array([], dtype=int64),) 0
ratio_pattern_len_to_pattern_4 (array([], dtype=int64),) 0
ratio_pattern_len_to_pattern_bad_len (array([], dtype=int64),) 0
last_microloan_openeddt (array([], dtype=int64),) 0
is_type_credit_card_100 (array([], dtype=int64),) 0
is_type_consumer_100 (array([], dtype=int64),) 0
is_type_micro_100 (array([], dtype=int64),) 0
is_active_type_credit_card_100 (array([], dtype=int64),) 0
is_active_type_consumer_100 (array([], dtype=int64),) 0
is_active_type_micro_100 (array([], dtype=int64),) 0
is_type_credit_card_12 (array([], dtype=int64),) 0
is_type_consumer_12 (array([], dtype=int64),) 0
is_type_micro_12 (array([], dtype=int64),) 0
is_active_type_credit_card_12 (array([], dtype=int64),) 0
is_active_type_consumer_12 (array([], dtype=int64),) 0
is_active_type_micro_12 (array([], dtype=int64),) 0
is_type_credit_card_3 (array([], dtype=int64),) 0
```

```
is_type_consumer_3 (array([], dtype=int64),) 0
is_type_micro_3 (array([], dtype=int64),) 0
is_active_type_credit_card_3 (array([], dtype=int64),) 0
is_active_type_consumer_3 (array([], dtype=int64),) 0
is_active_type_micro_3 (array([], dtype=int64),) 0
is_type_credit_card_1 (array([], dtype=int64),) 0
is_type_consumer_1 (array([], dtype=int64),) 0
is_type_micro_1 (array([], dtype=int64),) 0
is_active_type_credit_card_1 (array([], dtype=int64),) 0
is_active_type_consumer_1 (array([], dtype=int64),) 0
is_active_type_micro_1 (array([], dtype=int64),) 0
overall_worst_overdue_state_12 (array([], dtype=int64),) 0
ratio_sum_outstanding_to_open_sum (array([ 2646,  3078,  3644,  4839,  65
15,  7849, 11264, 11285, 11829,
       14644, 14993, 15045, 15311, 15711, 19327, 20002, 20956, 21193,
       21485, 21686, 23204, 23312, 23534, 23724, 24965, 25794, 27152,
       27178, 27963, 28574, 28646, 28698, 28990, 29996, 30095, 32165,
       32352, 32611, 32849, 33393, 33400, 33693, 34682, 35126, 35396,
       35878, 37382, 38213, 38735, 41499, 41584, 41724, 41879, 42247],
      dtype=int64),) 54
target (array([], dtype=int64),) 0
```

In [31]:

```
# меняю np.inf на медиану
cr = fillna_inf(cr)
```

baseline

In [32]:

```
x_train, x_test, y_train, y_test = train_test_split(cr.drop("target", axis=1), cr["targe
```

In [33]:

```
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
```

```
(34023, 74)
(34023,)
(8506, 74)
(8506,)
```

In [34]:

```
model = xgb.XGBClassifier(random_state=1)
model.fit(x_train, y_train)
```

Out[34]:

```
XGBClassifier(base_score=0.5, booster='gbtree', callbacks=None,
              colsample_bylevel=1, colsample_bynode=1, colsample_bytree=
1,
              early_stopping_rounds=None, enable_categorical=False,
              eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwis
e',
              importance_type=None, interaction_constraints='',
              learning_rate=0.300000012, max_bin=256, max_cat_to_onehot=
4,
              max_delta_step=0, max_depth=6, max_leaves=0, min_child_weig
ht=1,
              missing=nan, monotone_constraints='()', n_estimators=100,
              n_jobs=0, num_parallel_tree=1, predictor='auto', random_sta
te=1,
              reg_alpha=0, reg_lambda=1, ...)
```

In [35]:

```
pred = model.predict(x_test)
```

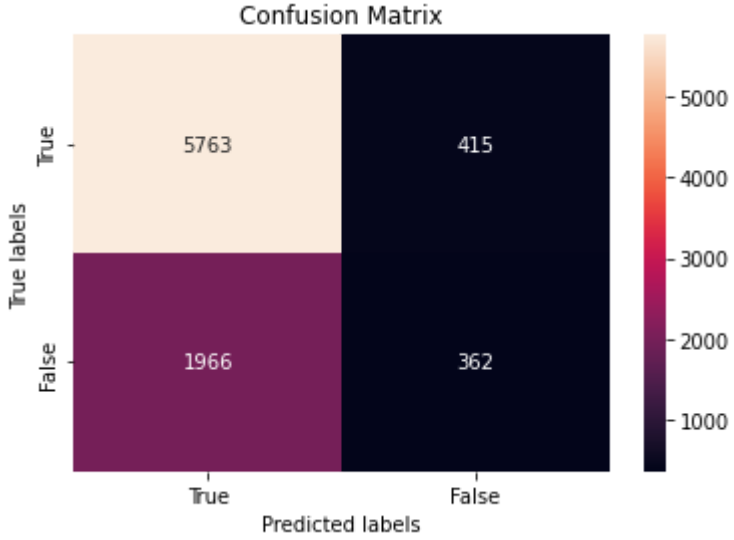
In [36]:

```
test_score = confusion_matrix(y_test, pred)
```

In [37]:

```
ax= plt.subplot()
sns.heatmap(test_score, annot=True, fmt='g', ax=ax);  #annot=True to annotate cells, ftm

# labels, title and ticks
ax.set_xlabel('Predicted labels');ax.set_ylabel('True labels');
ax.set_title('Confusion Matrix');
ax.xaxis.set_ticklabels(['True', 'False']); ax.yaxis.set_ticklabels(['True', 'False']);
```



In [38]:

```
print(classification_report(y_test, pred))
```

```
              precision    recall  f1-score   support

         0.0       0.75      0.93      0.83      6178
         1.0       0.47      0.16      0.23      2328

    accuracy                           0.72      8506
   macro avg       0.61      0.54      0.53      8506
weighted avg       0.67      0.72      0.67      8506
```

неплохой результат для baseline

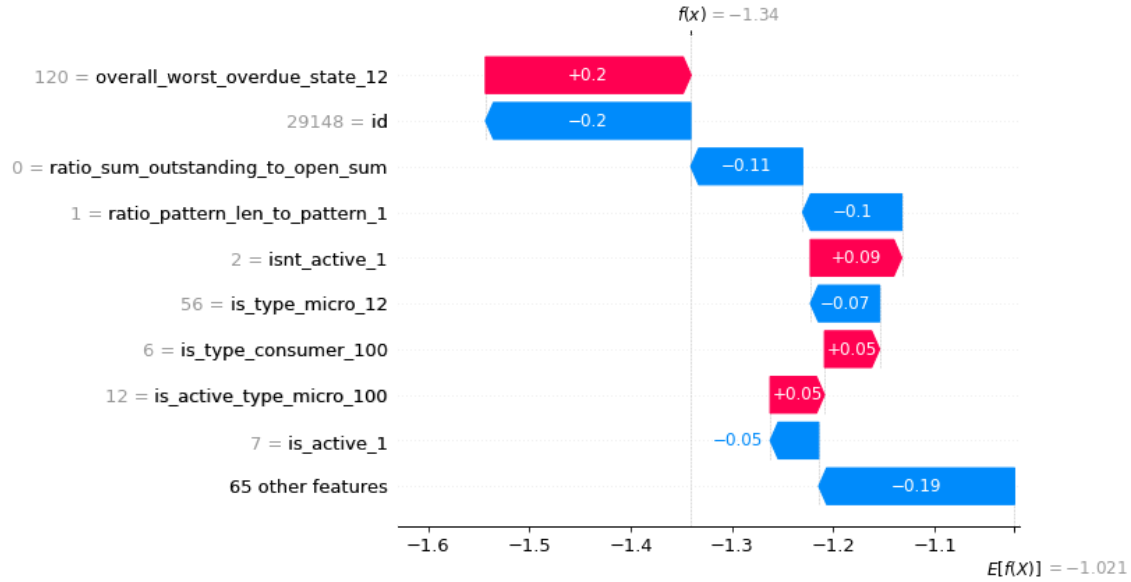
In [39]:

```
explainer = shap.Explainer(model, algorithm="tree")
shap_values = explainer(x_train)
```

```
ntree_limit is deprecated, use `iteration_range` or model slicing instea
d.
```

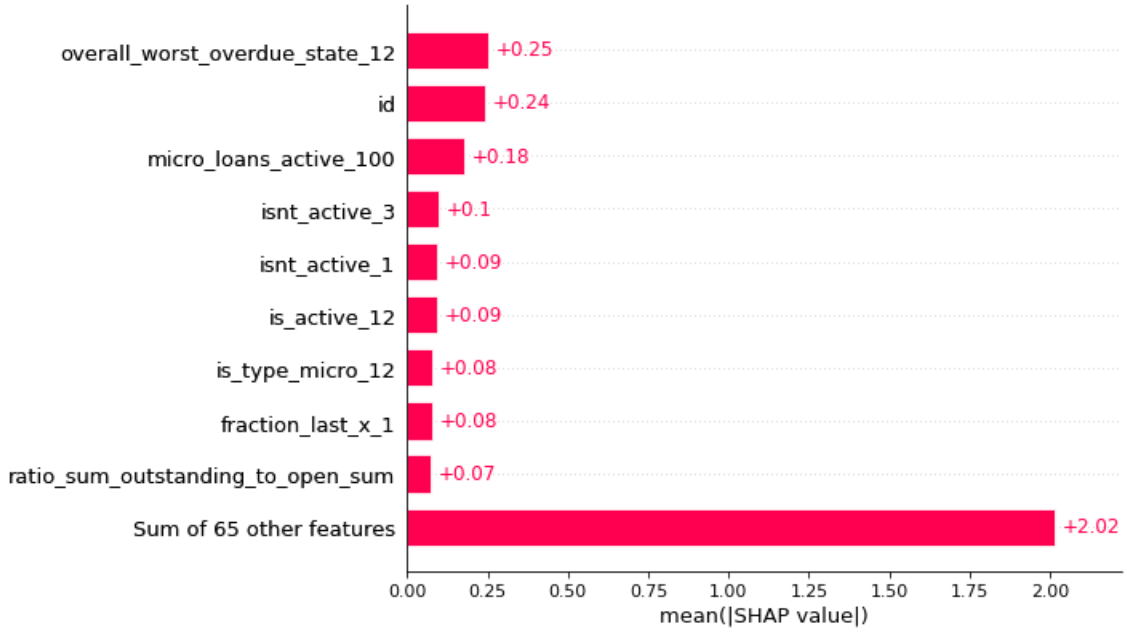
In [40]:

```
shap.plots.waterfall(shap_values[0])
```



In [41]:

```
shap.plots.bar(shap_values)
```



На графиках можно увидеть наиболее значимые признаки

посмотрим распределение целевой переменной

In [42]:

```
cr.target.value_counts()
```

Out[42]:

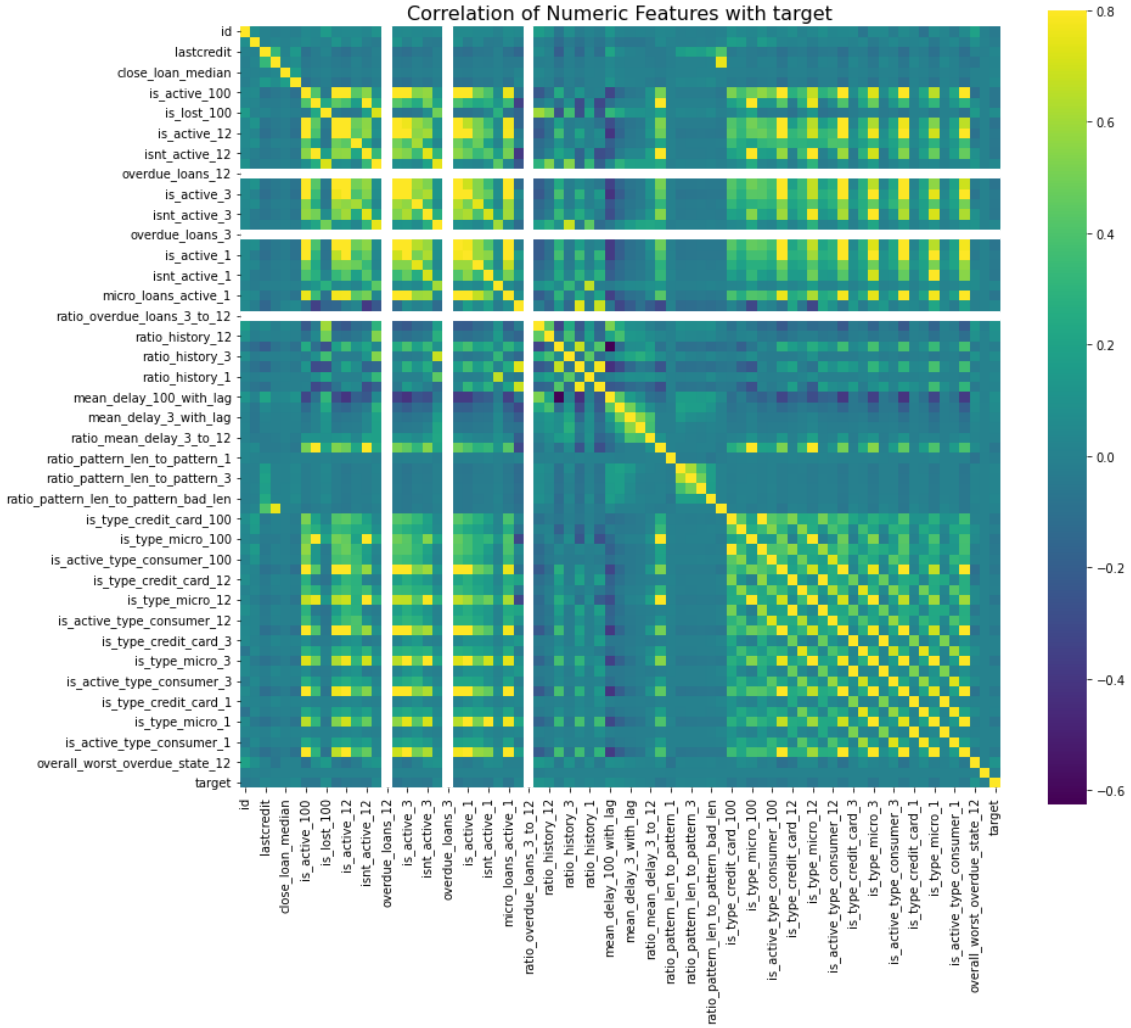
```
0.00000    30729
1.00000    11800
Name: target, dtype: int64
```

посмотрим линейную зависимость данных

In [43]:

```
list_corr, corr = cor(cr, "target")
```

```
target                            1.00000
overall_worst_overdue_state_12    0.10446
ratio_history_100                 0.10419
ratio_history_12                  0.08024
mean_delay_100_with_lag           0.07500
                                     ...
micro_loans_active_12            -0.07797
micro_loans_active_100           -0.08276
overdue_loans_12                      NaN
overdue_loans_3                       NaN
ratio_overdue_loans_3_to_12           NaN
Name: target, Length: 75, dtype: float64
```



Краткий вывод:
Видно, что некоторые признаки сильно коррелируют между собой, что для обучения модели будет не очень хорошо
Лучше от них избавиться на данном этапе

In [44]:

```
list_corr = dict(list_corr)
list_corr
```

Out[44]:

```
{'target': 1.0,
 'overall_worst_overdue_state_12': 0.10445678996592271,
 'ratio_history_100': 0.10418744916795468,
 'ratio_history_12': 0.08024497459465531,
 'mean_delay_100_with_lag': 0.07500454052382886,
 'is_lost_100': 0.06131998970117157,
 'ratio_history_3': 0.054237302599499694,
 'mean_delay_12_with_lag': 0.053987407871768735,
 'open_loan_median': 0.04106627856197758,
 'is_lost_12': 0.04047975379988168,
 'is_lost_3': 0.03633040390462604,
 'mean_delay_3_with_lag': 0.03188488534155772,
 'ratio_history_1': 0.0297240730939936,
 'ratio_pattern_len_to_pattern_bad_len': 0.02687464295209383,
 'is_lost_1': 0.02670666750656655,
 'lastcredit': 0.026328202312190217,
 'ratio_all_microloans_3_to_12': 0.02611643022024763,
 'ratio_pattern_len_to_pattern_2': 0.020506145839237547,
 'close_loan_median': 0.01989512137121537,
 'ratio_pattern_len_to_pattern_3': 0.019444614337089383,
 'ratio_pattern_len_to_pattern_4': 0.017390374144382367,
 'mean_delay_1_with_lag': 0.015902800000541263,
 'last_microloan_openeddt': 0.014773067589946372,
 'is_type_consumer_1': 0.013081021686110533,
 'time_to_lastcredit_closeddt': 0.012339446462624 86,
 'fraction_last_x_1': 0.010583162807648115,
 'is_type_consumer_3': 0.009144324633819312,
 'ratio_mean_delay_3_to_12': 0.008592402694227518,
 'is_active_type_consumer_1': 0.0079106671213074,
 'is_active_type_consumer_3': 0.006604653274036468,
 'is_active_type_credit_card_3': 0.006355316390335842,
 'is_active_type_credit_card_1': 0.005779611754077081,
 'is_type_credit_card_1': 0.005410817399518365,
 'fraction_last_x_3': 0.005284952641682 5126,
 'is_type_credit_card_3': 0.005159591247837 23,
 'is_type_consumer_12': [illegible],
 'ratio_sum_outstanding_to_open_sum': -0.008155259937400405,
 'is_active_type_consumer_12': [illegible],
 'is_active_type_credit_card_12': [illegible],
 'is_type_consumer_100': [illegible],
 'is_type_credit_card_12': [illegible],
 'fraction_last_x_12': [illegible],
 'ratio_pattern_len_to_pattern_1': [illegible],
 'age': [illegible],
 'is_active_type_consumer_100': [illegible],
 'is_active_1': [illegible],
 'is_active_type_micro_1': [illegible],
 'is_active_3': [illegible],
 'is_active_type_micro_3': [illegible],
 'id': [illegible],
 'open_sum_1': [illegible],
 'is_active_type_credit_card_100': -0.04687678987765133 6,
 'open_sum_3': [illegible],
 'is_type_credit_card_100': [illegible],
 'is_active_12': -0.0496828776142308 1,
 'is_active_type_micro_12': -0.05031709578439231,
 'isnt_active_3': -0.05206561984150432,
 'is_active_type_micro_100': -0.054588700570409904,
 'open_sum_12': -0.05566151154928737,
 'isnt_active_1': -0.05684408668858893,
 'isnt_active_100': -0.05763505898992622,
```

In [45]:

```python
for k, v in list_corr.items():
    if abs(v) > 0.1:
        print(k,v)
```

```
target 1.0
overall_worst_overdue_state_12 0.10445678996592271
ratio_history_100 0.10418744916795468
```

краткий вывод: Линейная зависимость у данных с целевой переменной слабая

In [46]:

```python
signs_corr = check_signs_corr(corr)
signs_corr
```

```
{[illegible]
```

In [47]:

```
len(cr.columns)
```

Out[47]:

75

In [48]:

```
cr_c = cr.copy()
```

In [49]:

```
# Удаляю признаки коррелирующие между собой
cr_c = del_columns_corr(cr, signs_corr)
```

In [50]:

```
len(cr_c.columns)
```

Out[50]:

24

In [51]:

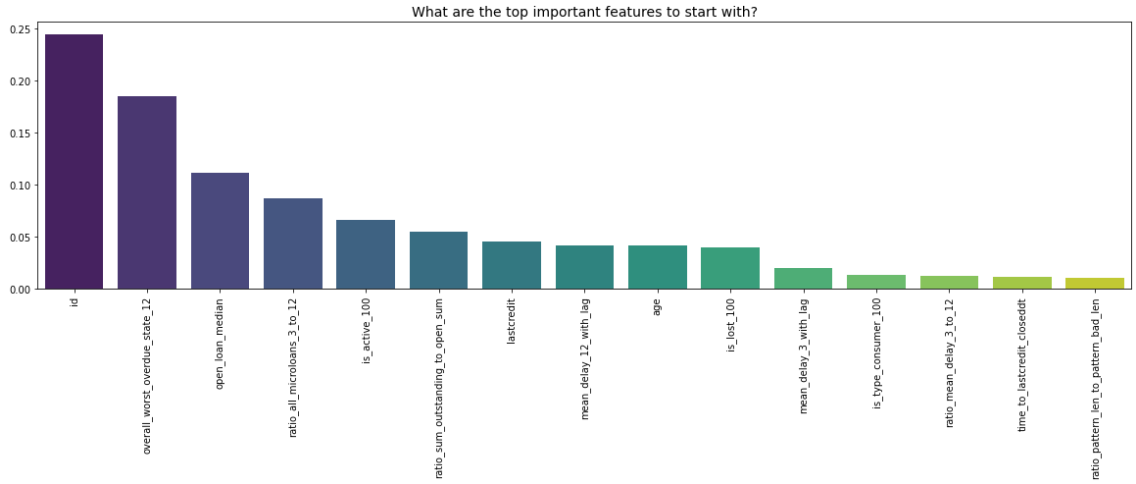
```
chech_omissions(cr_c)
```

```
всего строк: 42529
id: кол-во пропусков: 0, процент: 0 %
age: кол-во пропусков: 0, процент: 0 %
lastcredit: кол-во пропусков: 0, процент: 0 %
time_to_lastcredit_closeddt: кол-во пропусков: 0, процент: 0 %
close_loan_median: кол-во пропусков: 0, процент: 0 %
open_loan_median: кол-во пропусков: 0, процент: 0 %
is_active_100: кол-во пропусков: 0, процент: 0 %
is_lost_100: кол-во пропусков: 0, процент: 0 %
overdue_loans_12: кол-во пропусков: 0, процент: 0 %
overdue_loans_3: кол-во пропусков: 0, процент: 0 %
ratio_all_microloans_3_to_12: кол-во пропусков: 0, процент: 0 %
ratio_overdue_loans_3_to_12: кол-во пропусков: 0, процент: 0 %
mean_delay_12_with_lag: кол-во пропусков: 0, процент: 0 %
mean_delay_3_with_lag: кол-во пропусков: 0, процент: 0 %
ratio_mean_delay_3_to_12: кол-во пропусков: 0, процент: 0 %
ratio_pattern_len_to_pattern_1: кол-во пропусков: 0, процент: 0 %
ratio_pattern_len_to_pattern_2: кол-во пропусков: 0, процент: 0 %
ratio_pattern_len_to_pattern_4: кол-во пропусков: 0, процент: 0 %
ratio_pattern_len_to_pattern_bad_len: кол-во пропусков: 0, процент: 0 %
is_type_consumer_100: кол-во пропусков: 0, процент: 0 %
is_type_credit_card_3: кол-во пропусков: 0, процент: 0 %
overall_worst_overdue_state_12: кол-во пропусков: 0, процент: 0 %
ratio_sum_outstanding_to_open_sum: кол-во пропусков: 0, процент: 0 %
target: кол-во пропусков: 0, процент: 0 %
```

найдем нелинейную зависимость признаков с целевой переменной

In [52]:

```
nonlianer = nonlianer_comun(cr_c, "target")
```



In [53]:

```
len(nonlianer[0])
```

Out[53]:

15

Краткий вывод:
14 наиболее значимых признаков зависимых нелинейно
Думаю будет достаточно для начала и получения хорошего baseline
ТК зависимости нелинейные, буду использовать нелинейные модели

In [54]:

```
nonlianer = list(nonlianer[0])
```

In [55]:

```
nonlianer.append("target")
```

In [56]:

```
nonlianer
```

Out[56]:

```
['id',
 'overall_worst_overdue_state_12',
 'open_loan_median',
 'ratio_all_microloans_3_to_12',
 'is_active_100',
 'ratio_sum_outstanding_to_open_sum',
 'lastcredit',
 'mean_delay_12_with_lag',
 'age',
 'is_lost_100',
 'mean_delay_3_with_lag',
 'is_type_consumer_100',
 'ratio_mean_delay_3_to_12',
 'time_to_lastcredit_closeddt',
 'ratio_pattern_len_to_pattern_bad_len',
 'target']
```

In [57]:

```
cr_c = cr_c[nonlianer]
```

In [58]:

```
cr_c.shape
```

Out[58]:

```
(42529, 16)
```

In [59]:

```
cr_c.drop("id", inplace=True, axis=1)
```

train model

XGBClassifier

In [60]:

```
model_fit(xgb.XGBClassifier, cr_c)
```

```
x_train.shape:  (34023, 14)
y_train.shape:  (34023,)
x_test.shape:  (8506, 14)
y_test.shape:  (8506,)
              precision    recall  f1-score   support

         0.0       0.73      0.96      0.83      6140
         1.0       0.47      0.08      0.14      2366

    accuracy                           0.72      8506
   macro avg       0.60      0.52      0.48      8506
weighted avg       0.66      0.72      0.64      8506
```

Out[60]:

```
(XGBClassifier(base_score=0.5, booster='gbtree', callbacks=None,
               colsample_bylevel=1, colsample_bynode=1, colsample_bytr
ee=1,
               early_stopping_rounds=None, enable_categorical=False,
```

Type *Markdown* and LaTeX: α^2

In [61]:

```
from sklearn import preprocessing
```

In [62]:

```
X = cr_c.drop("target", axis=1)
```

In [63]:

```
x = X.values #returns a numpy array
min_max_scaler = preprocessing.MinMaxScaler()
x_scaled = min_max_scaler.fit_transform(x)
df = pd.DataFrame(x_scaled, columns=[X.columns])
```

In [64]:

```
df["target"] = cr_c["target"]
```

In [65]:

```
model_xgb, x_train, x_test = model_fit(xgb.XGBClassifier, df)
```

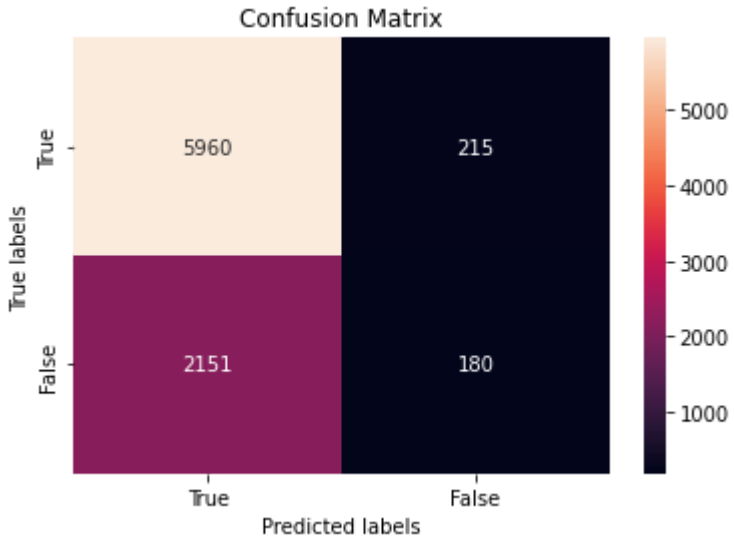
```
x_train.shape:  (34023, 14)
y_train.shape:  (34023, 1)
x_test.shape:  (8506, 14)
y_test.shape:  (8506, 1)

dropping on a non-lexsorted multi-index without a level parameter may imp
act performance.

              precision    recall  f1-score   support

         0.0       0.73      0.97      0.83      6175
         1.0       0.46      0.08      0.13      2331

    accuracy                           0.72      8506
   macro avg       0.60      0.52      0.48      8506
weighted avg       0.66      0.72      0.64      8506
```



catboost

In [66]:

```
from catboost import CatBoostClassifier
```

In [67]:

```
model_cat, x_train, x_test = model_fit(CatBoostClassifier, df)
```

```
x_train.shape:  (34023, 14)
y_train.shape:  (34023, 1)
x_test.shape:  (8506, 14)
y_test.shape:  (8506, 1)
Learning rate set to 0.046452

dropping on a non-lexsorted multi-index without a level parameter may
impact performance.

0:      learn: 0.6828558        total: 189ms    remaining: 3m 8s
1:      learn: 0.6729944        total: 217ms    remaining: 1m 48s
2:      learn: 0.6640291        total: 242ms    remaining: 1m 20s
3:      learn: 0.6558172        total: 264ms    remaining: 1m 5s
4:      learn: 0.6484386        total: 286ms    remaining: 56.8s
5:      learn: 0.6418077        total: 305ms    remaining: 50.6s
6:      learn: 0.6357257        total: 325ms    remaining: 46s
7:      learn: 0.6306305        total: 342ms    remaining: 42.4s
8:      learn: 0.6259841        total: 357ms    remaining: 39.3s
9:      learn: 0.6213834        total: 372ms    remaining: 36.8s
10:     learn: 0.6173608        total: 387ms    remaining: 34.8s
```

Краткий вывод:
Поиск нелинейных зависимостей и удаление коррелирующих между собой признаков только ухудшили модель. Можно пойти по пути постепенного удаления признаков, начиная с тех, которые менее важны. Для начала, конечно, лучше выбрать модель для дальнейшего обучения.
Для улучшеня признакого пространства можно использовать алоритмы уменьшения размерности.
Создать собственные признаки. Например, отношение возраста к возращенным кредитам за весь срок или за месяц и тд.
Посмотреть, какой с ними будет результат.
Хооший Результат дает ансамбль моделей.
Сделать доверительный интервал.
В создании модели хорошо иметь такие признаки, как: наличие работы, наличие машины или квартиры, семейное положение, наличие детей. Если есть возможность, то узнать, на, что берутся деньги.