Машинное обучение для предсказания дефолта по кредиту

Итоговая работа по курсу «Deep Learning»

Специализация «Data Scientist»

Малинина Елена
Группа DSU - 1

Постановка задачи

Машинное обучение (Machine Learning, ML)

- форма искусственного интеллекта (ИИ), которая позволяет системе итеративно обучаться на данных, используя различные алгоритмы для описания данных и прогнозирования результатов.
В данной работе рассматривается метод прогнозирование кредитоспособности клиентов банка Home Credit

Актуальность задачи

Рост спроса на кредитование ведет к увеличению конкуренции на рынке и необходимость обработки большого количества данных. Такие условия обязывают почти полностью автоматизировать процессы с наименьшим участием человека.

Подобные процессы автоматизации также требуют гибкой настройки, адаптации, обучения на новых данных и построения более совершенных моделей.

Описание проблемы

Кредитный скоринг представляет собой математическую или статистическую модель, с помощью которой на основе кредитной истории «прошлых» клиентов банк пытается определить, насколько велика вероятность, что конкретный потенциальный заемщик вернет кредит в срок.

Повышение доходности кредитных операций непосредственно связано с качеством оценки кредитного риска. В зависимости от классификации клиента по группам риска банк принимает решение, стоит ли выдавать кредит или нет, какой лимит кредитования и проценты следует устанавливать.

Основная задача – **оценка риска**

Этапы решения задачи

1. Первичный анализ данных (EDA) и поиск аномалий
2. Оценка корреляции значений
3. Кодирование категориальных значений
4. Заполнение пропущенных значений
5. Конструирование признаков
6. Обучение логистической регрессии
7. Обучение CatBoostClassifier (модель градиентного бустинга)
8. Вывод

Целевая метрика

Рассматриваемые нами метрики основаны на использовании следующих исходов:

- TP - истинно-положительное решение
- TN - истинно-отрицательное решение
- FP - ложно-положительное решение
- FN - ложно-отрицательное решение

Тогда, точность и полнота определяются следующим образом:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

F-мера представляет собой гармоническое среднее между точностью и полнотой.

$$F = 2\frac{Precision \times Recall}{Precision + Recall}$$

Анализ данных

Exploratory Data Analysis (EDA)

2

Был использован датасет, содержащий информацию о клиентах банка Home Credit
Ноутбук доступен по ссылке

- **Объем датасета:**
  307511 строк. Одна строка - представляет один кредит. В датасете указаны характеристики заемщика, такие как пол, возраст, место работы, семейный статус, уровень дохода, наличие недвижимости и машины и т.д, всего 122
  Взяты также несколько дополнительных датасетов. Общий датасет составил 207 колонок.

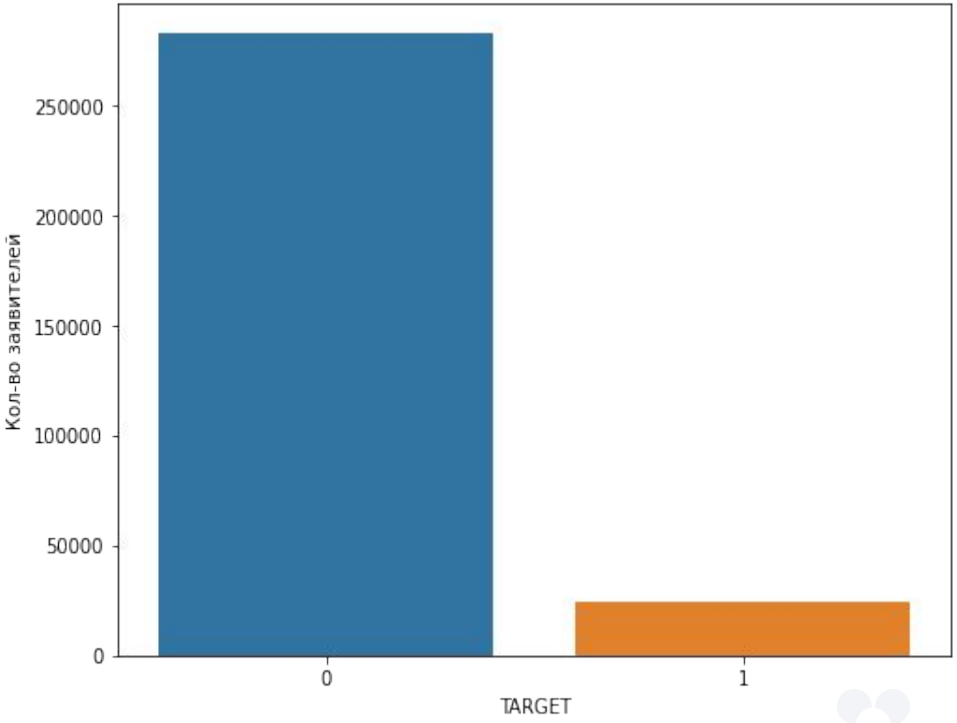
- **Target:**
  является целевой бинарной переменной, где 0 - кредит погашен, 1 - кредит не погашен

Целевая переменная и распределение

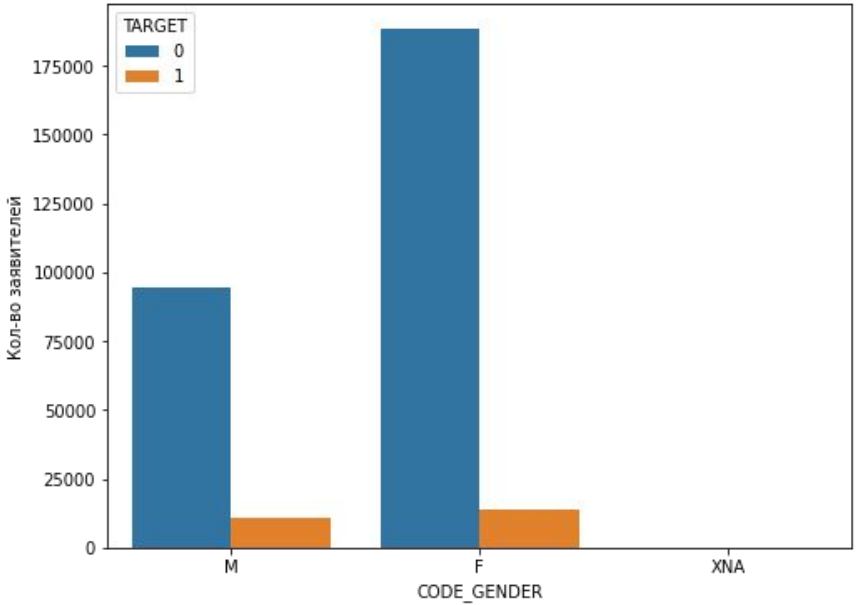
```
1 target_count = df_train['TARGET'].value_counts()
2 print('Класс 0:', target_count[0])
3 print('Класс 1:', target_count[1])
```

```
Класс 0: 282686
Класс 1: 24825
```

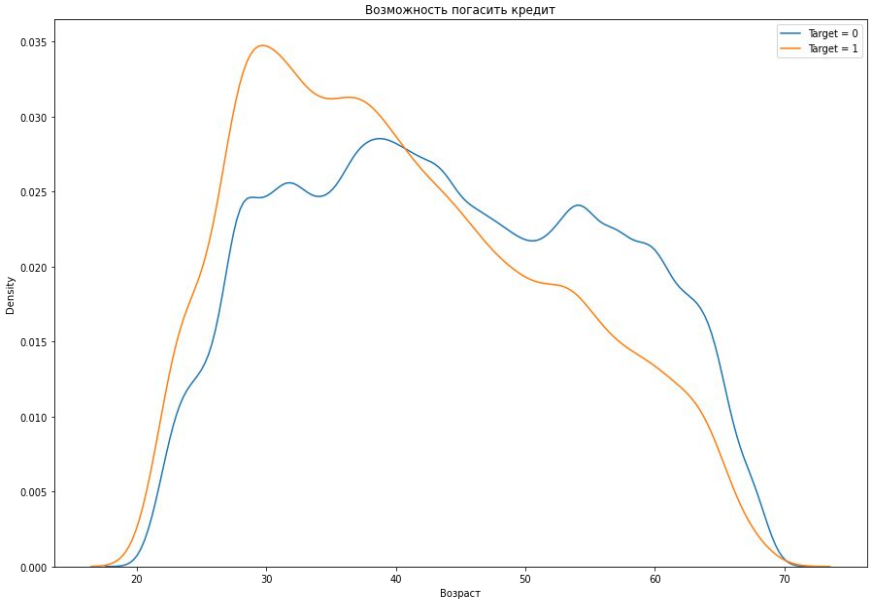
Данные не сбалансированы. Это приведет к неправильному прогнозу в пользу 0-го класса



Поиск аномалий 1.0

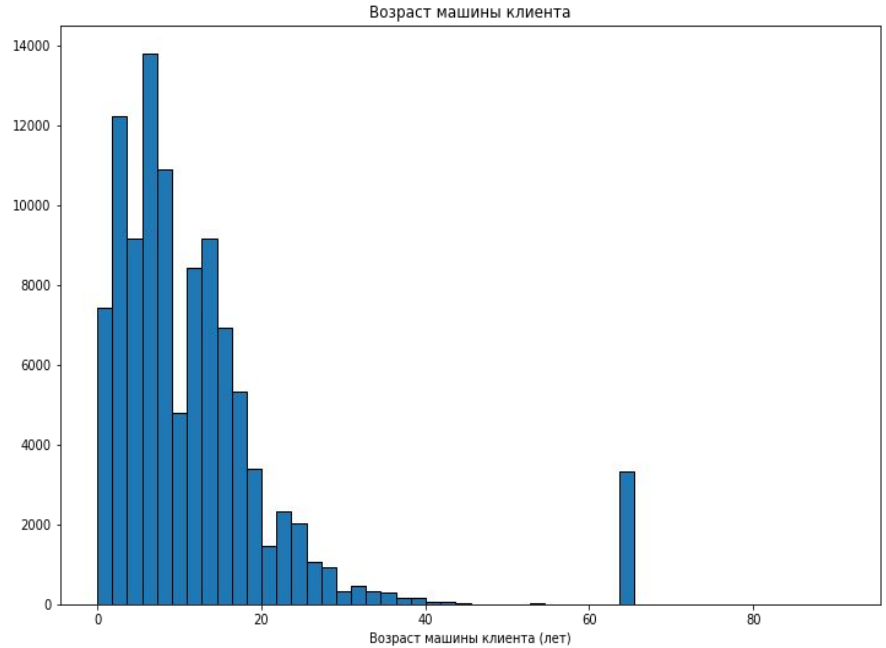


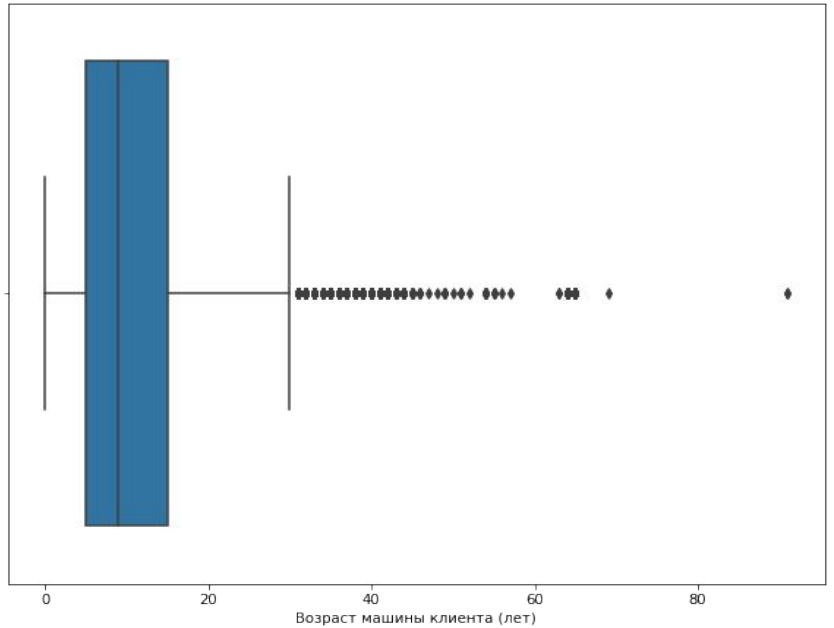
Рассматривается пол заемщика и чаще всего это женщины. Судя по графику проблемы с выплатой у женщин почти наравне с мужчинами, хотя мужчины берут кредиты реже. Также присутствует неизвестный класс XNA, значений в нем всего 4 шт. и их можно убрать.



Разброс возраста от 20 до 70 лет, всплеск кредитования приходится в диапазоне от 25 - 45 лет. Можно увидеть, что проблемы с выплатой приходится примерно на тот же период жизни. При этом, чем старше человек, тем меньше вероятность не погасить кредит.

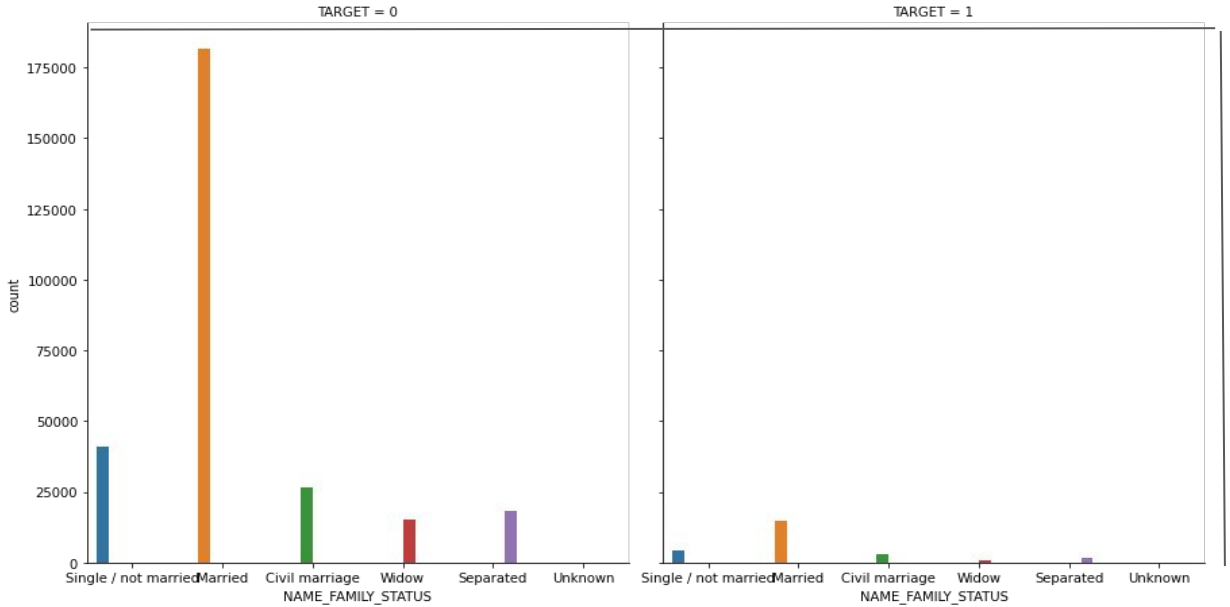
Поиск аномалий 1.1





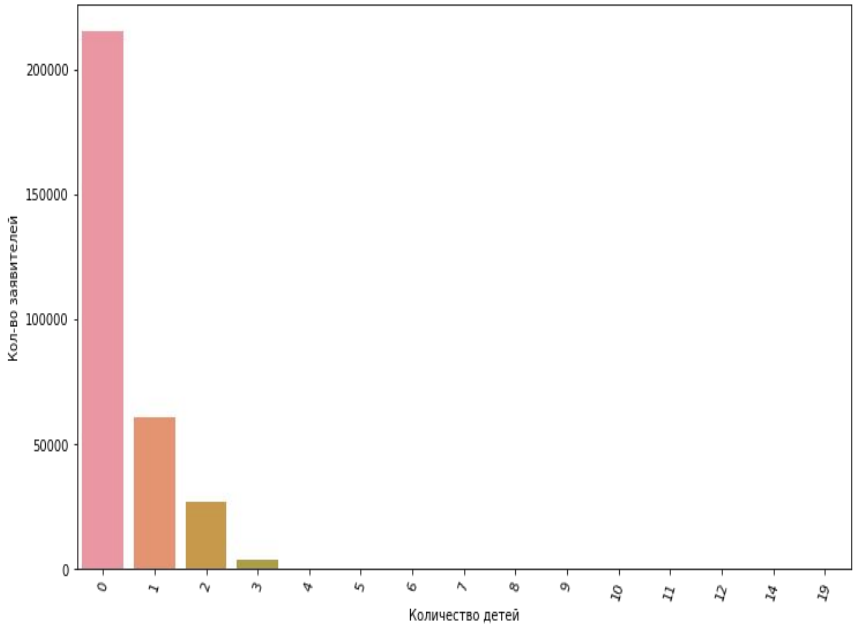
Возраст машины клиента имеет выбросы - их уберем

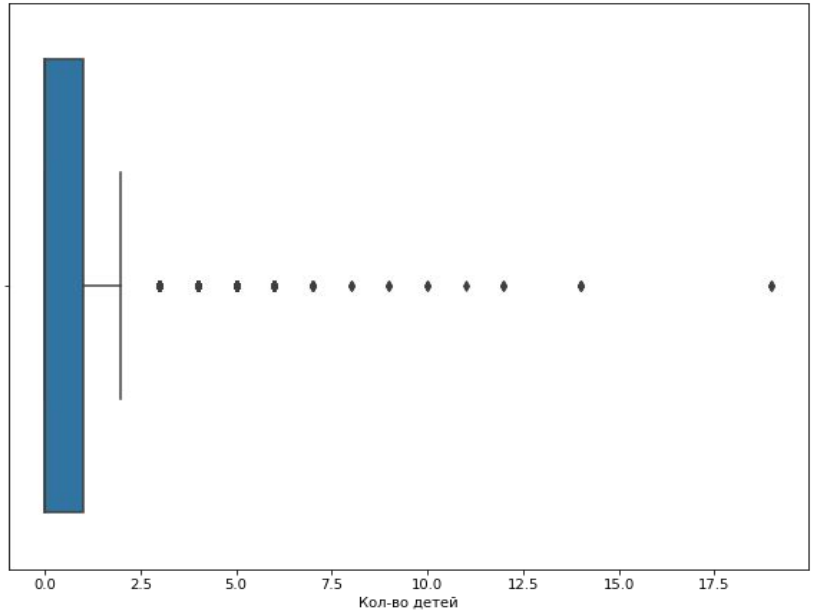
Поиск аномалий 1.3



Семейный статус клиента может быть весомым при прогнозе, его тоже рассмотрим. Большинство женаты / замужем. Есть класс Unknown, там представлены 2 строки, его убираем.

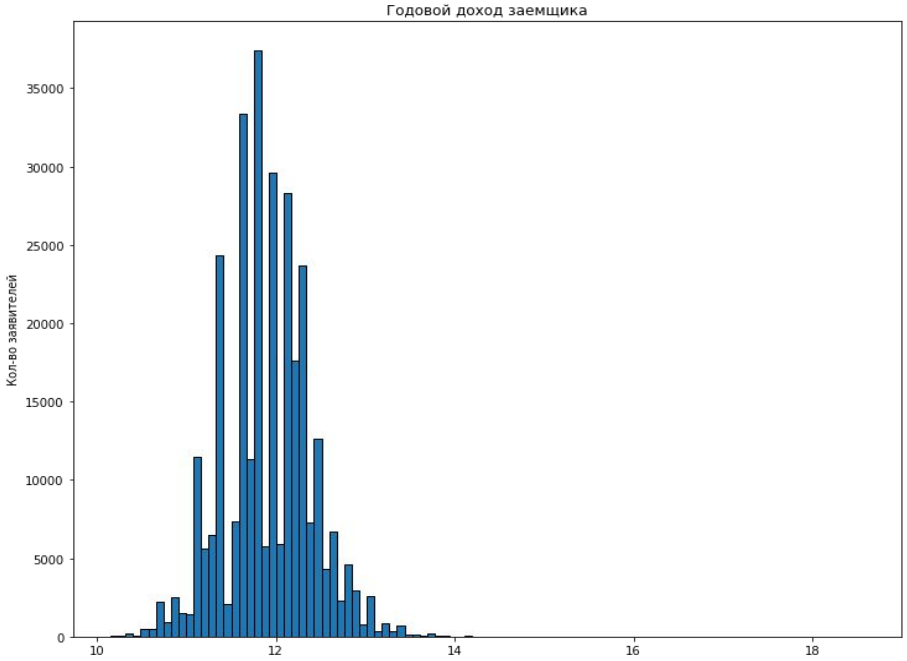
Поиск аномалий 1.4

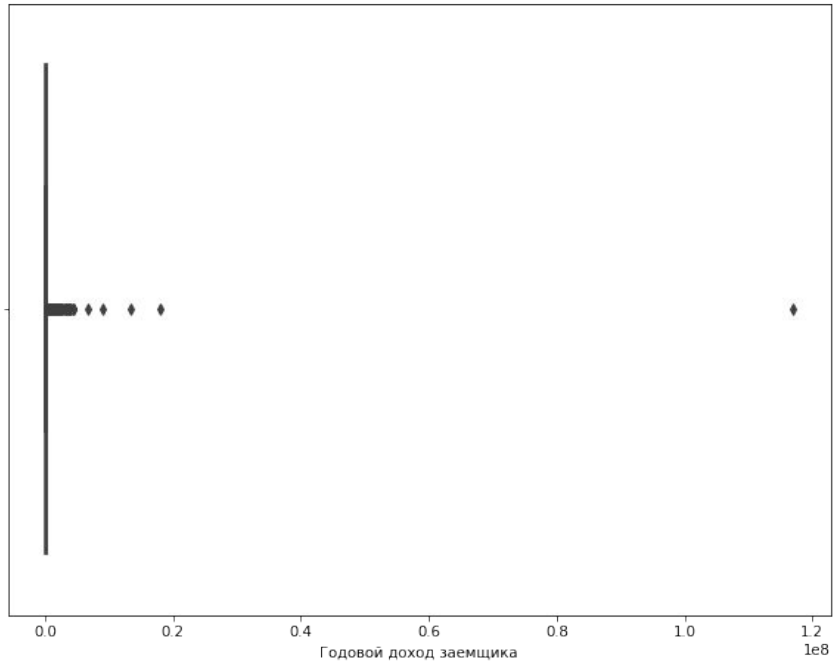




Большинство детей не имеют. На boxplot есть отметки, где больше 10 детей. Это вполне реальная ситуация ,но этих данных всего 8 строк, их тоже уберем, так как может быть перекос в данных.

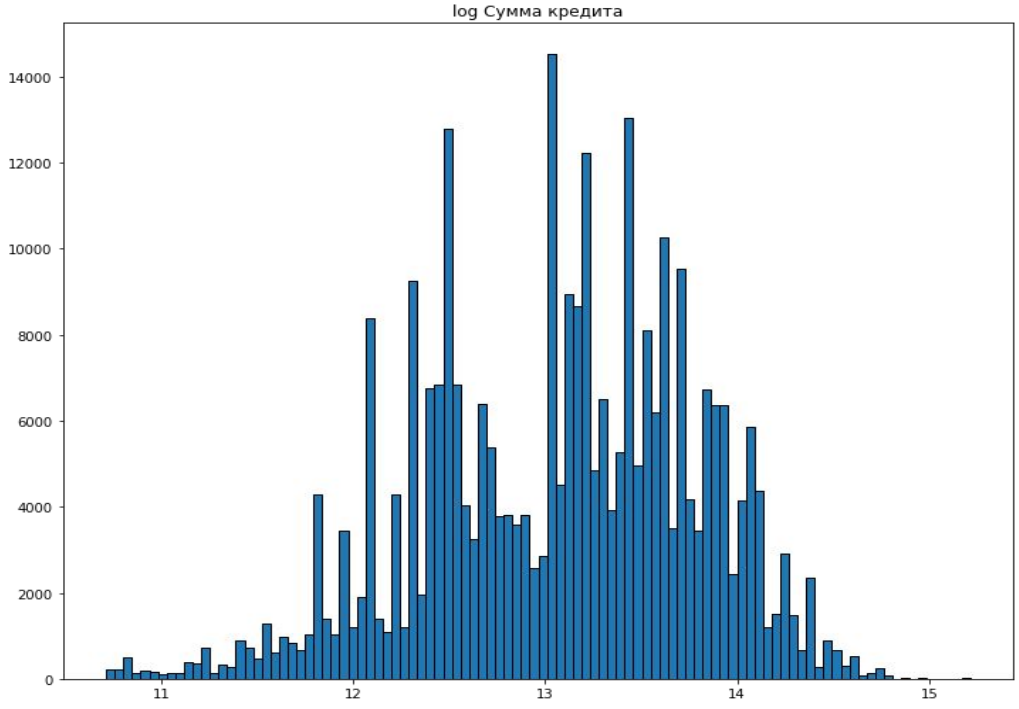
Поиск аномалий 1.5





В boxplot годовом доходе заемщика есть значения превышающие 12 000 000. Их всего 3 шт. Эти данные лучше убрать из - за большого разрыва.

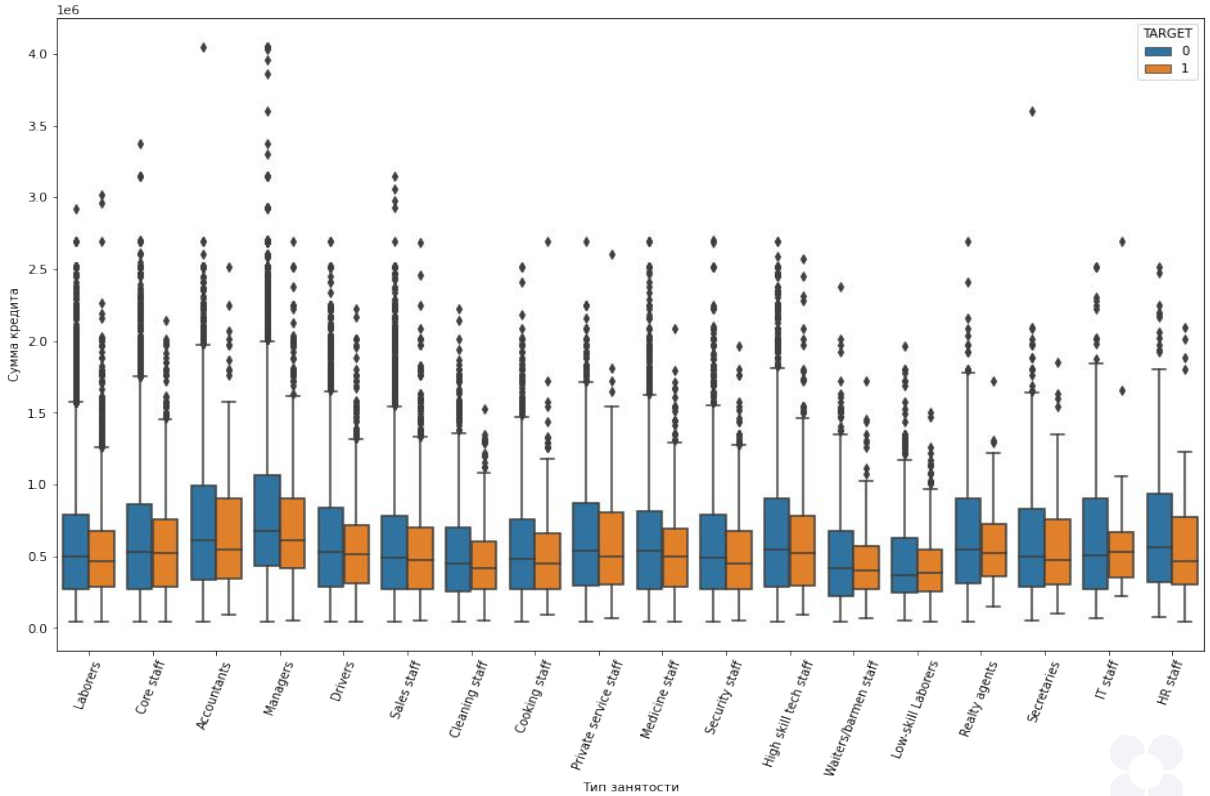
Поиск аномалий 1.6



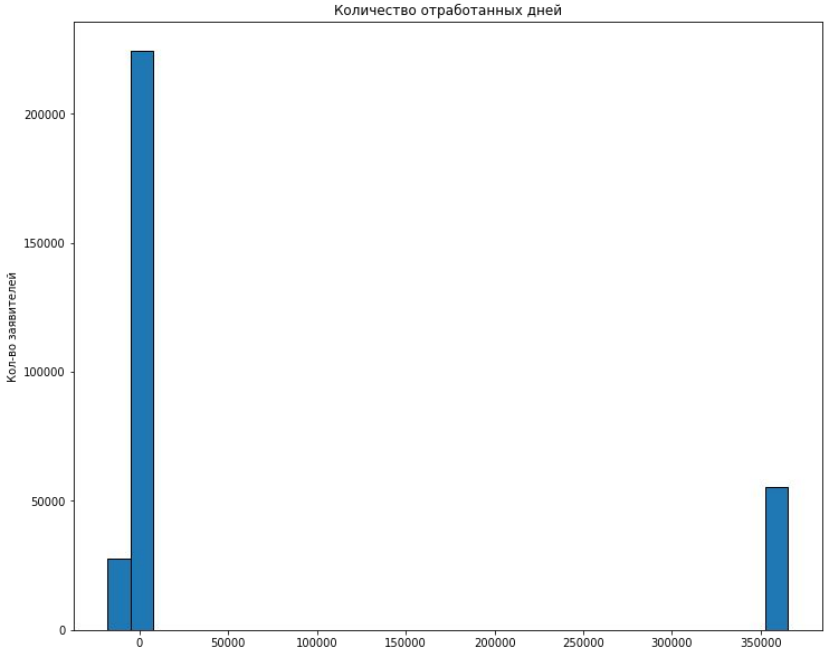
Кредиты на сумму > 400000 брали коммерческие партнеры в основном

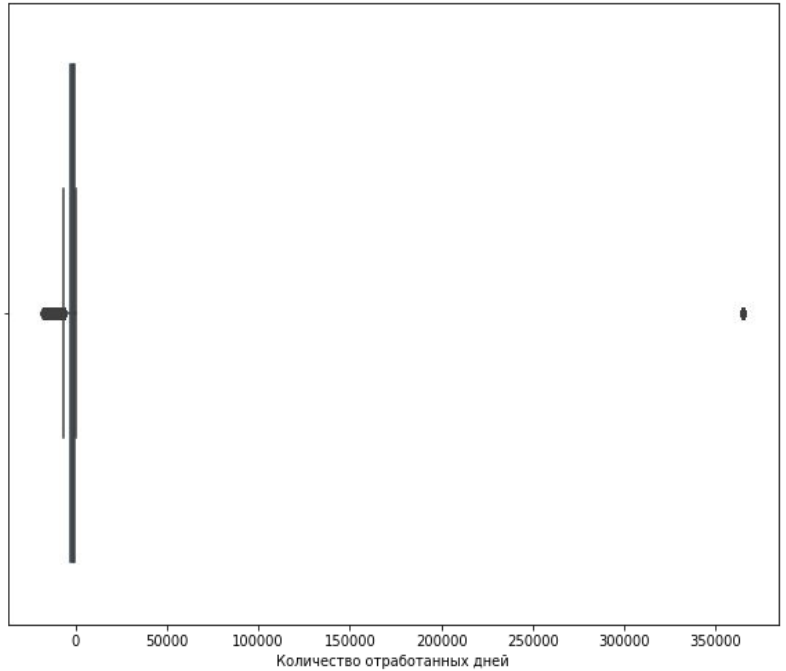
Поиск аномалий 1.7

Отношение типа занятости и суммы кредита, где люди чаще имеют задолженности.
Здесь присутствуют большие выбросы - их отсечем



Поиск аномалий 1.8



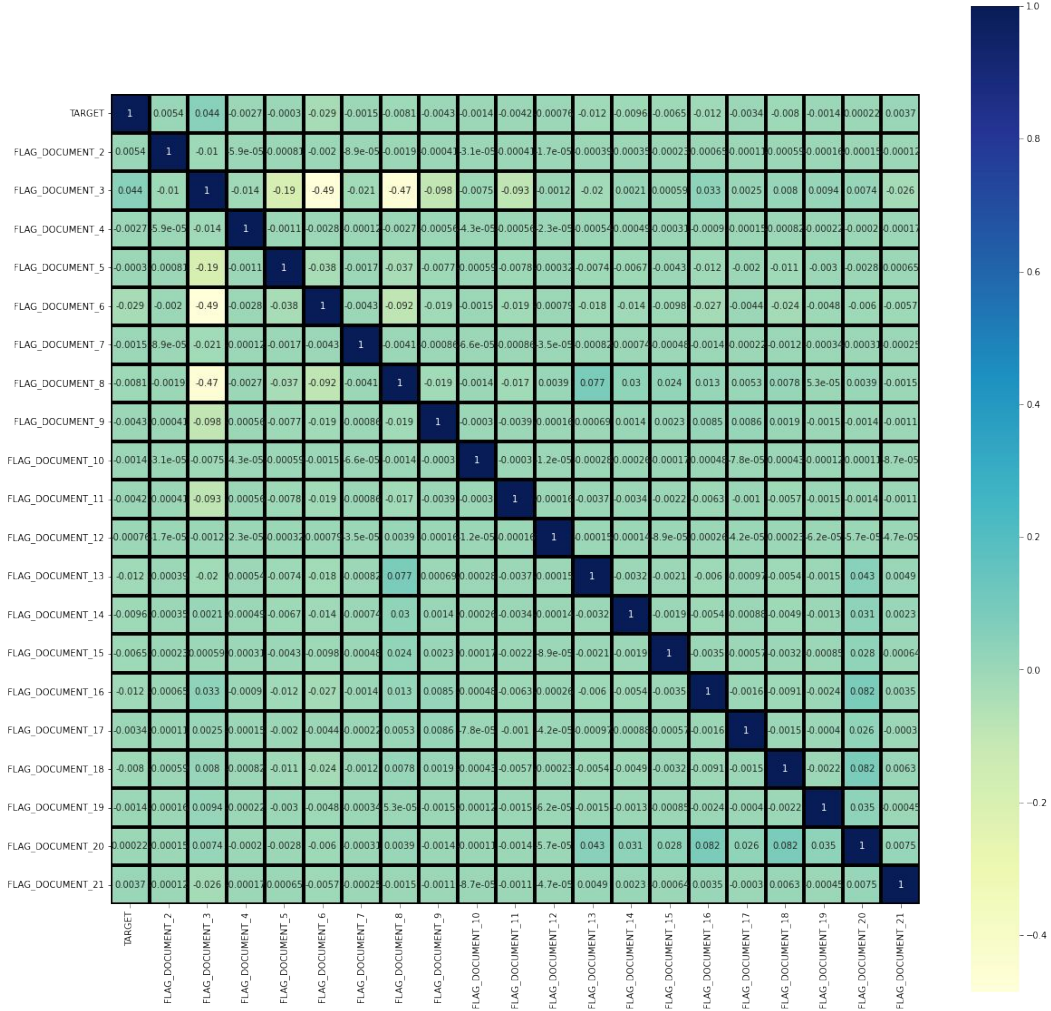


Взглянем на количество отработанных дней. Есть значение в 365243 дня, оно равняется 1000 лет. Эти данные составляют 18% от всего дата фрейма, это достаточно много, поэтому эти значения заменяем на нули.

Матрица корреляции

Среди предоставленных документов клиентом банку есть значительные зависимости, отберем самые значения с самой низкой корреляцией, а именно FLAG_DOCUMENT_6, FLAG_DOCUMENT_13, FLAG_DOCUMENT_16, FLAG_DOCUMENT_14,

FLAG_DOCUMENT_20



Кодирование категориальных переменных

Всего в датасете 16 категориальных переменных. Будем использовать стандартную библиотеку из Scikit Learn - LabelEncoder

- NAME_CONTRACT_TYPE
- CODE_GENDER
- FLAG_OWN_CAR
- FLAG_OWN_REALTY
- NAME_TYPE_SUITE
- NAME_INCOME_TYPE
- NAME_EDUCATION_TYP
- NAME_FAMILY_STATUS
- NAME_HOUSING_TYPE
- OCCUPATION_TYPE
- WEEKDAY_APPR_PROCESS_START
- ORGANIZATION_TYPE
- FONDKAPREMONT_MODE
- HOUSETYPE_MODE
- WALLSMATERIAL_MODE
- EMERGENCYSTATE_MODE

Обработка пропущенных значений

Пропущенных данных много. Справа показаны первые 20 колонок, где значений Nan больше всего. Воспользуемся Simple Imputer из библиотеки Scikit Learn, заполним переменные модой.

	0
COMMONAREA_AVG	214858
COMMONAREA_MEDI	214858
COMMONAREA_MODE	214858
NONLIVINGAPARTMENTS_MODE	213509
NONLIVINGAPARTMENTS_AVG	213509
NONLIVINGAPARTMENTS_MEDI	213509
LIVINGAPARTMENTS_MODE	210194
LIVINGAPARTMENTS_MEDI	210194
LIVINGAPARTMENTS_AVG	210194
FLOORSMIN_MEDI	208636
FLOORSMIN_MODE	208636
FLOORSMIN_AVG	208636
YEARS_BUILD_MEDI	204483
YEARS_BUILD_MODE	204483
YEARS_BUILD_AVG	204483
OWN_CAR_AGE	202925
LANDAREA_AVG	182584
LANDAREA_MEDI	182584
LANDAREA_MODE	182584
BASEMENTAREA_AVG	179939

Конструирование признаков (Feature Engineering)

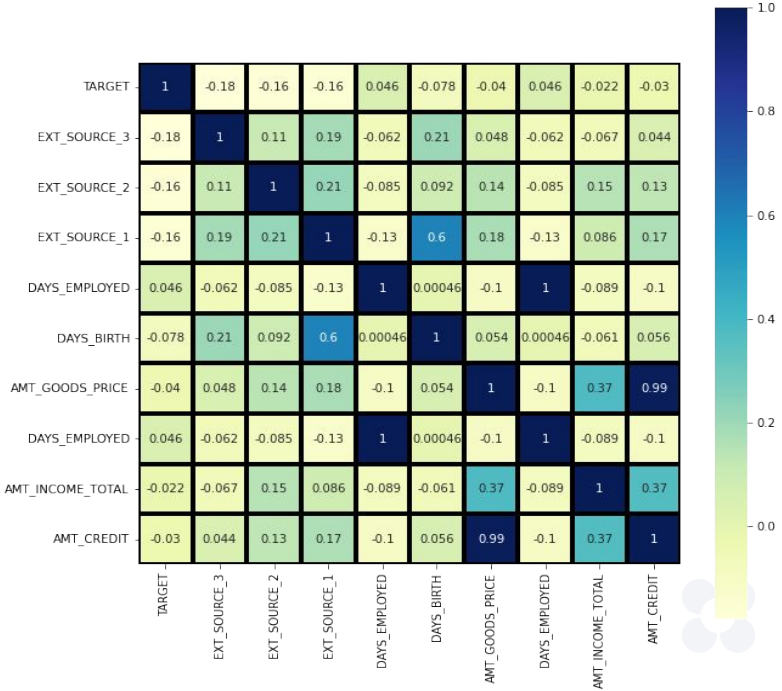
Уберем ID клиента

```
1 # сразу убрем ID клиента (он не нужен)
2 df_train = df_train.drop('SK_ID_CURR', axis=1)
3 df_train.shape
```

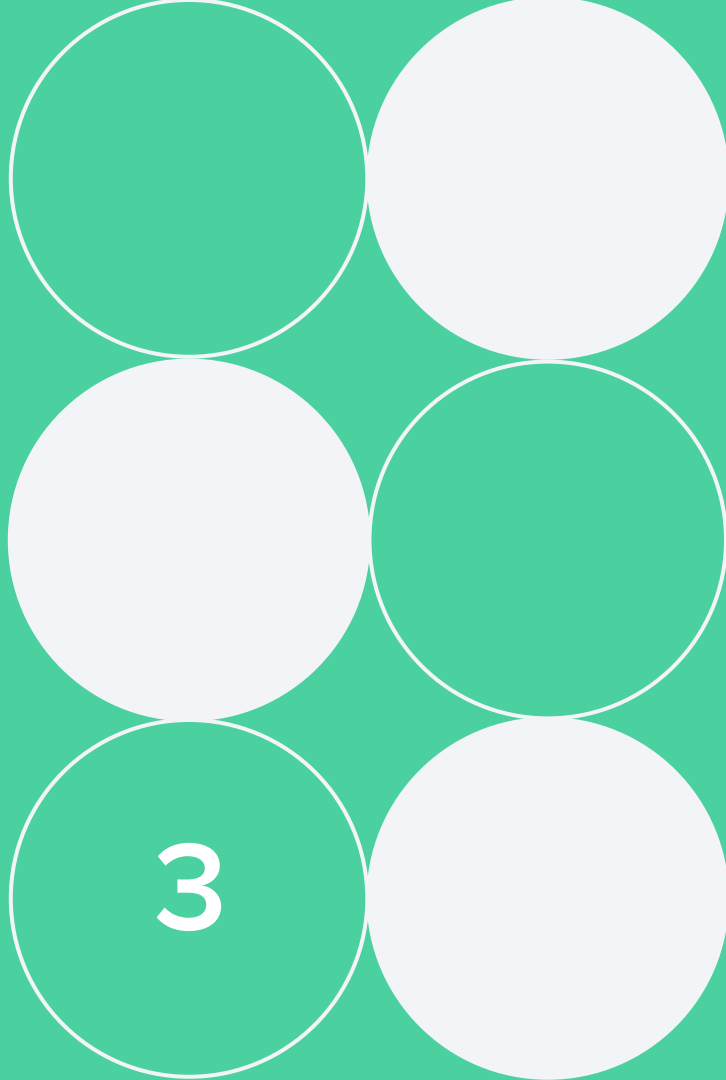
```
(307479, 106)
```

И посмотрим на корреляцию, самых важных на первый взгляд, признаков. Создадим несколько фич для разрыва зависимости между данными

- RATIO_CREDIT_PRICE_% - отношение кредита и стоимости покупок
- RATIO_CREDIT_ANNUITY_% - представление о сроках кредита
- RATIO_GOODS_INCOME_% - отношение зарплаты и стоимости покупок



Обучение алгоритмов



LogisticRegression. Метрики

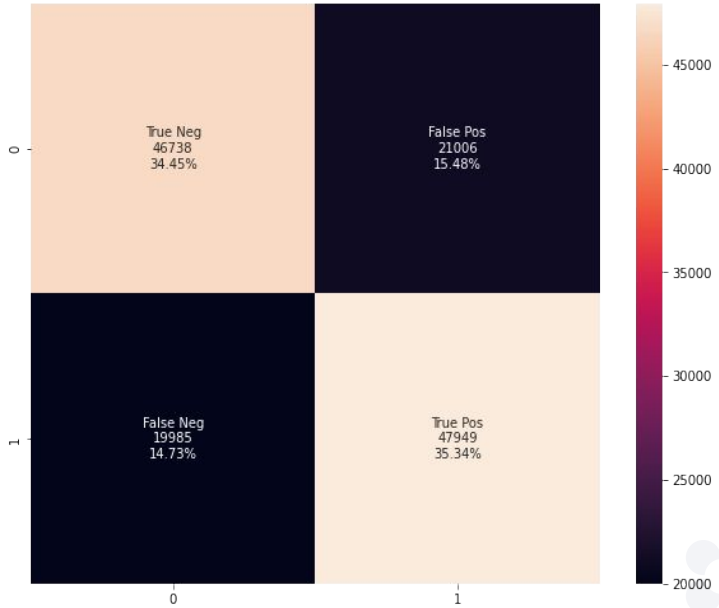
Для начала нормализуем данные с помощью MinMaxScaler и сбалансируем целевую переменную используя библиотеку SMOTE, где будем дублировать минорный класс. Делим данные (20% - тестовая часть). Так же посмотрим на матрицу ошибок

```
Качество модели на test: 0.6978802753578325
Качество модели на train: 0.6971264857966839
```

```
Accuracy - 0.6961
Recall - 0.7023
Precision - 0.6943
ROC AUG - 0.6961
              precision    recall  f1-score   support

           0       0.70      0.69      0.69     67744
           1       0.69      0.70      0.70     67934

    accuracy                           0.70    135678
   macro avg       0.70      0.70      0.70    135678
weighted avg       0.70      0.70      0.70    135678
```



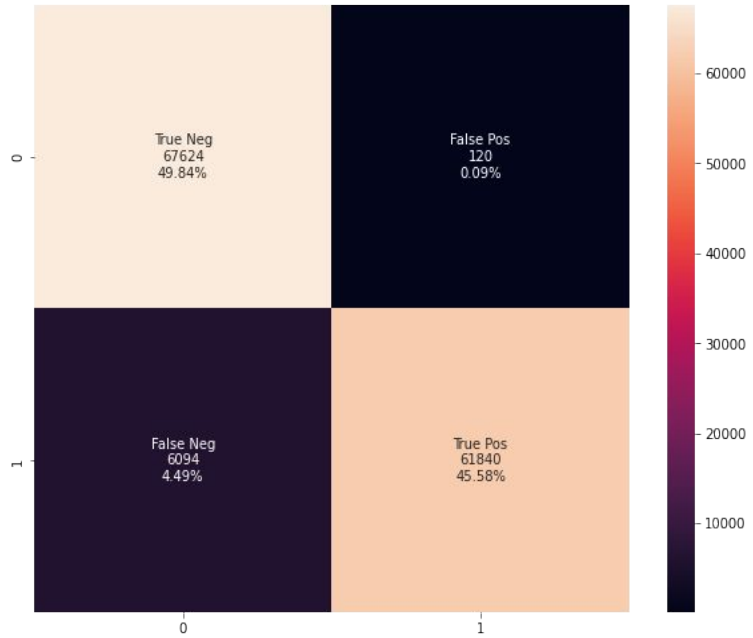
CatBoostClassifier. Метрики

```
Качество модели на test: 0.9542003862085231
Качество модели на train: 0.9543623906677912
```

```
Accuracy - 0.9542
Recall - 0.9103
Precision - 0.9981
ROC AUG - 0.9543
              precision    recall  f1-score   support

           0       0.92      1.00      0.96     67744
           1       1.00      0.91      0.95     67934

    accuracy                           0.95    135678
   macro avg       0.96      0.95      0.95    135678
weighted avg       0.96      0.95      0.95    135678
```



Итоги обучения

4

LogisticRegression. Результат

- Модель логистической регрессии не смогла показать хороший результат в решении этой задачи, точность в районе 70%.

TRAIN	precision	recall	f1-score	support
0	0.70	0.69	0.70	226366
1	0.69	0.70	0.70	225890
accuracy			0.70	452256
macro avg	0.70	0.70	0.70	452256
weighted avg	0.70	0.70	0.70	452256

TEST	precision	recall	f1-score	support
0	0.70	0.70	0.70	56294
1	0.70	0.70	0.70	56770
accuracy			0.70	113064
macro avg	0.70	0.70	0.70	113064
weighted avg	0.70	0.70	0.70	113064

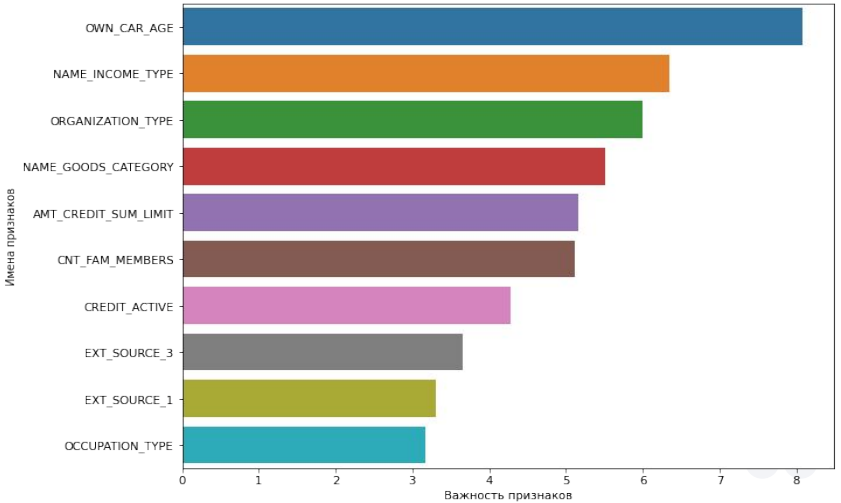
CatBoostClassifier. Результат

- CatBoostClassifier показал хороший результат, переобучения нет

TRAIN	precision	recall	f1-score	support
0	0.92	1.00	0.96	226299
1	1.00	0.91	0.95	225960
accuracy			0.95	452259
macro avg	0.96	0.95	0.95	452259
weighted avg	0.96	0.95	0.95	452259

TEST	precision	recall	f1-score	support
0	0.92	1.00	0.96	56363
1	1.00	0.91	0.95	56702
accuracy			0.96	113065
macro avg	0.96	0.96	0.95	113065
weighted avg	0.96	0.96	0.95	113065

- На целевую переменную влияет:
  - OWN_CAR_AGE (кол-во лет машине владельца)
  - NAME_INCOME_TYPE (тип дохода клиента)
  - ORGANIZATION_TYPE (организация, где работает клиент)



Выводы

5

1. Исходя из этой работы можно сказать что, полученные результаты демонстрируют, что данная задача успешно решается методами классического машинного обучения

2. Модель CatBoost лучше подходит для данной задачи, чем LogisticRegression

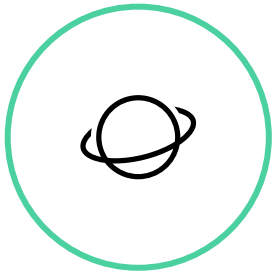
3. Из - за несбалансированной таргетной переменной нужно создавать синтетические данные

4. Гипотеза: возможно, добавление новых признаков (как анализ соц.сетей, например) повысит качество моделей

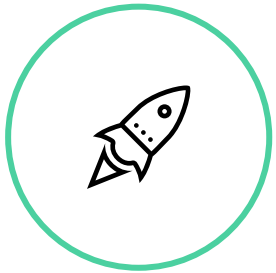
Заключение

6

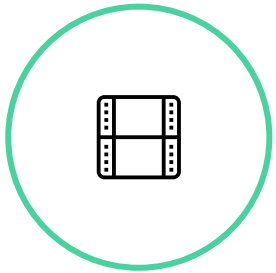
3 элемента



Произведен анализ исходных данных, выявлены и устранены значительные аномалии



Произведено сравнение результатов модели логистической регрессии и градиентного бустинга



Реализована модель ML, которая выявила определенные закономерности

Спасибо за внимание!