

Московский государственный университет имени М.В. Ломоносова

Направление 38.03.01 Экономика

Программа «Национальная экономика»

**Прогнозирование рисков банковского кредитования с  
использованием технологий искусственного  
интеллекта**

Выпускная квалификационная работа

Студент

Касенова Асем Ардаковна

Научный руководитель:

к.э.н., к.ф.-м.н., к.ю.н., доцент

Сидоренко Владимир Николаевич

Астана

2026

# ОГЛАВЛЕНИЕ

Глава 1. Алгоритмы машинного и глубокого обучения в задаче о кредитном скро-	
ринге . . . . .	3
1.1 Предобработка данных кредитной истории клиентов банка . . . . .	3
1.1.1 Метод главных компонент . . . . .	5
1.1.2 Определение аномального клиента . . . . .	8
1.1.3 Статистический тест Колмогорова-Смирнова . . . . .	12
1.2 Базовые алгоритмы оценки вероятности возврата кредита . . . . .	15
1.2.1 Деревья решений . . . . .	15
1.2.2 Случайный лес . . . . .	15
1.2.3 Градиентный бустинг . . . . .	15
1.2.4 Нейронные сети . . . . .	15
Приложение 1. Предобработка данных . . . . .	17

# ГЛАВА 1. АЛГОРИТМЫ МАШИННОГО И ГЛУБОКОГО ОБУЧЕНИЯ В ЗАДАЧЕ О КРЕДИТНОМ СКОРИНГЕ

## 1.1 Предобработка данных кредитной истории клиентов банка

В данной работе используются обезличенные данные АО «Альфа-Банка»<sup>1</sup>. Данные состоят из 12 файлов (`train_data_0.pq` – `train_data_11.pq`), содержащих информацию о платежах клиентов банка. В каждом из 12 файлов содержится информация о 250 000 клиентах. При этом один клиент может иметь несколько кредитов, и каждому такому клиенту соответствует персональный `id` (идентификационный номер). Отдельно имеется файл `train_target.csv`, который состоит из 3 млн строк, и каждая строка соответствует клиенту с меткой (флагом) равной 0 (отсутствие дефолта) или 1 (наличие дефолта). Задача предобработки данных состоит в структурировании исходной информации, т.е. формирование единого датасета, выделение важных признаков (колонок), выявление аномальных клиентов (определение аномальности будет приведено ниже), визуализация данных и тестирование моделей МО и ГО на этих данных. Программный код формирования единого датасета реализован в листинге 1 (см. приложение 1).

**Комментарий 1.** Пояснение к листингу: 1

1. Строки 1 – 3. Импортируются необходимые библиотеки: `pandas` – для работы с табличными данными, `os` – для работы с файловой системой и `pyarrow.parquet` – для чтения файлов формата `.parquet`.
2. Стока 5. Задается путь `path = "train_data"` к папке, в которой находятся исходные файлы формата `.pq`.
3. Строки 6 – 13. Запускается цикл `for`, который перебирает все файлы в папке `train_data`. Формируется имя поочередного файла `train_data_i.pq`, создается объект `ParquetDataset` для текущего файла, из которого данныечитываются в `DataFrame` (`df`). Затем выполняется агрегация данных по признаку `id`, вычисляются средние значения признаков, после чего данные сохраняются в соответствующий `csv` – файл в папку `train_data_csv_all`.
4. Стока 14. Выводится список файлов каталога `train_data` для проверки корректности формирования файлов.
5. Строки 16 – 21. Задается путь к папке с полученными `csv`-файлами `train_data_csv_all`. Создается пустой список `frames` для последующего хранения отдельных `DataFrame`.

<sup>1</sup>Соревнование на данных кредитных историй [Электронный ресурс] / Open Data Science. URL: <https://ods.ai/competitions/dl-fintech-bki> (дата обращения 25.11.2025)

Затем запускается цикл по файлам в папке `train_data_csv_all`, который последовательно перебирает все файлы в формате `.csv`. Результат сохраняется в `DataFrame` (`df`) и добавляется в список `frames`.

6. Строки 23 – 26. Все элементы списка `frames` объединяются в единый `DataFrame` `result` с помощью функции `pd.concat`. Полученный датасет сохраняется в файл `1_data_csv_all.csv`, после чего заново считывается в переменную `df_all` для последующего анализа.

После преобразования получился следующий датасет. В таблице 1 приводится фрагмент датасета:

Таблица 1 – Фрагмент преобразованного датасета, содержащий первых 5 клиентов

<code>id</code>	<code>enc_paym_0</code>	<code>enc_paym_1</code>	<code>enc_paym_2</code>	<code>enc_paym_3</code>	<code>enc_paym_4</code>	<code>flag</code>
1750000	0.17	0.17	0.17	0.33	0.67	0
1750001	0.00	0.75	0.75	0.75	0.75	0
1750002	0.39	0.33	0.72	0.67	1.17	0
1750003	0.18	0.23	0.41	0.50	0.55	0
1750004	0.60	0.60	0.60	0.60	1.20	0

Примечание: `id` – идентификатор заявки; `enc_paym_0`, ..., `enc_paym_n` – статусы ежемесячных платежей за последние  $n$  месяцев; `flag` – статус кредита (0=кредит полностью оплачен). Полный датасет состоит из 61 признака, включая дополнительные характеристики по кредитам.

## Комментарий 2. Пояснение к агрегированию клиентов в листинге: 1

Агрегация клиентов по идентификатору `id` в строке 11 необходима для определения итоговой метки (флага) каждого клиента, поскольку одному заемщику может соответствовать несколько кредитных договоров. Задача состоит в присвоении каждому кредиту одного заемщика единую итоговую метку. В данном исследовании в качестве общего значения для всех кредитов используется среднее исходных значений. Для понимания идеи приводится следующий пример в виде таблицы 2 :

Таблица 2 – Дисциплина оплаты кредитов клиента (`id = 1`)

<code>id</code>	<code>N</code>	<code>M1</code>	<code>M2</code>	<code>M3</code>	<code>M4</code>	<code>M5</code>	<code>M6</code>	<code>flag</code>	
1	1	1	0	1	1	0	1	0	
1	2	0	1	0	1	1	2		
1	3	1	2	0	0	3	2		
		Среднее значение							
1		0.67	1	0.33	1	1.33	1.67	0	

Источник: составлено автором на основе: Соревнование на данных кредитных историй [Электронный ресурс] / Open Data Science. – URL: <https://ods.ai/competitions/dl-fintech-bki> (дата обращения: 25.11.2025).

В данной таблице признаки означают следующее:

1. `id` – идентификационный номер клиента;
2. `N` – номер кредита;
3.  $M_1, \dots, M_6$  – статусы погашения в течение 6 месяцев (0=платеж вовремя оплачен, 1=задержка платежа на 1 день, 2=задержка платежа на 2 дня, 3=задержка платежа на 3 дня);
4. `flag` – статус кредита (0=кредит полностью оплачен).

В таблице 2 на пересечении  $N=2$  (второго кредита) и  $M_2$  (платеж во втором месяце) выделена цифра 1, означающая, что платеж по второму кредиту во втором месяце был оплачен с опозданием на 1 день.

Таким образом, данные трех строк, соответствующих трем кредитам клиента, в таблице 2 были усреднены и преобразованы в одну строку, которая содержит агрегированную информацию по клиенту с `id = 1`. В целом агрегировать клиентов можно не только по среднему значению, но и по моде или медиане. Однако в данной работе выбрано среднее значение, поскольку оно обладает важными статистическими свойствами, такими как несмещенност и состоятельность. После группирования данных был сформирован новый датасет, состоящий из 3 млн клиентов, каждому из которых соответствует одна итоговая метка. Получившийся датасет содержит 61 признак, из которых далее необходимо отобрать наиболее информативные, т.е. такие признаки, существенно влияющие на точность алгоритмов МО и ГО.

### 1.1.1 Метод главных компонент

Метод главных компонент (англ. *principal component analysis, PCA*) – метод сокращения размерности данных, позволяющий уменьшать количество признаков с сохранением максимального объема исходной информации[PCA], на которых обучаются модели МО и ГО.

Как уже было отмечено выше сформированный датасет содержит 61 признак (столбец). Задача состоит в том, чтобы найти такие признаки, на которые модели МО и ГО показывали приемлемую точность. Следует отметить, что редукция признаков может уменьшить точность алгоритмов, поэтому необходимо внимательно следить за процессом сокращения данных. В качестве тестового алгоритма был выбран алгоритм *Random Forest* (случайный лес), поскольку в статье С.В. Смирнова[Smirnov] был проведен анализ предпочтения исследователей к алгоритмам МО, показавший, что наиболее популярным является случайный лес.

Из преобразованного датасета изначально выбираются только некатегориальные признаки и формируется новый датасет, содержащий 41 признак. Ниже приводится смысл этих признаков:

1. `pre_pterm` – плановое количество дней с даты открытия кредита до даты его закрытия;

2. `pre_fterm` – фактическое количество дней с даты открытия кредита до даты его закрытия;
3. `pre_loans_next_pay_summ` – сумма следующего платежа по кредиту;
4. `pre_loans_outstanding` – оставшаяся невыплаченная сумма кредита;
5. `pre_loans_total_overdue` – текущая просроченная задолженность по кредиту;
6. `pre_loans_max_overdue_sum` – максимальная просроченная задолженность по кредиту за весь срок;
7. `pre_loans_credit_cost_rate` – полная стоимость кредита;
8. `is_zero_loans5` – флаг: нет просрочек до 5 дней;
9. `is_zero_loans530` – флаг: нет просрочек от 5 до 30 дней;
10. `is_zero_loans3060` – флаг: нет просрочек от 30 до 60 дней;
11. `is_zero_loans6090` – флаг: нет просрочек от 60 до 90 дней;
12. `is_zero_loans90` – флаг: нет просрочек более чем на 90 дней;
13. `pre_util` – отношение оставшейся невыплаченной суммы кредита к кредитному лимиту;
14. `pre_maxover2limit` – отношение максимальной просроченной задолженности к кредитному лимиту;
15. `is_zero_util` – флаг: отношение оставшейся невыплаченной суммы кредита к кредитному лимиту равно 0;
16. `is_zero_over2limit` – флаг: отношение текущей просроченной задолженности к кредитному лимиту равно 0;
17. `enc_paym_0, ..., enc_paym_n` – статусы ежемесячных платежей за последние  $n$  месяцев.

Далее из этих 41 признака необходимо выбрать только такие, которые вносят наибольший вклад в информативность данных. В листинге 2 (см. Приложение 1) реализован метод главных компонент. На таком наборе данных алгоритм показывает следующие метрики (см. Таблицу 3):

Таблица 3 – Метрики точности на 41 признаке

Алгоритм – Случайный лес						
Flag метка	Precision точность	Recall полнота	f1-score f1-мера	Accuracy точность	ROC AUC	Количество
Тренировочная выборка						
0	1.00	0.97	0.98	0.97	0.51	724588
1	0.00	0.00	0.00			25412
Тестовая выборка						
0	1.00	0.97	0.98	0.97	0.50	241569
1	0.00	0.00	0.00			8431

Источник: составлено автором на основе: Соревнование на данных кредитных историй [Электронный ресурс] / Open Data Science. – URL: <https://ods.ai/competitions/dl-fintech-bki> (дата обращения: 25.11.2025).

Были получены результаты *PCA*:

$$\begin{aligned} \lambda_1^{(p)} &= 22.1, \lambda_2^{(p)} = 20.3, \lambda_3^{(p)} = 17.9, \lambda_4^{(p)} = 10.1, \lambda_5^{(p)} = 9.7, \\ \lambda_6^{(p)} &= 7.7, \lambda_7^{(p)} = 4.4, \lambda_8^{(p)} = 1.7, \lambda_9^{(p)} = 1.5, \lambda_{10}^{(p)} = 0.9, \\ \lambda_{11}^{(p)} &= 0.6, \lambda_{12}^{(p)} = 0.5, \lambda_{13}^{(p)} = 0.4, \lambda_{14}^{(p)} = 0.3, \lambda_{15}^{(p)} = 0.2, \\ \lambda_{16}^{(p)} &= 0.2, \lambda_{17}^{(p)} = 0.2, \lambda_{18}^{(p)} = 0.1, \lambda_{19}^{(p)} = 0.1, \lambda_{20}^{(p)} = 0.1, \\ \lambda_{21}^{(p)} &= 0.1, \lambda_{22}^{(p)} = 0.1, \lambda_{23}^{(p)} = 0.1, \lambda_{24}^{(p)} = 0.1, \lambda_{25}^{(p)} = 0.1, \\ \lambda_{26}^{(p)} &= 0.1, \lambda_{27}^{(p)} = 0.1, \lambda_{28}^{(p)} = 0.1, \lambda_{29}^{(p)} = 0.1, \lambda_{30}^{(p)} = 0.1, \\ \lambda_{31}^{(p)} &= 0.0, \lambda_{32}^{(p)} = 0.0, \lambda_{33}^{(p)} = 0.0, \lambda_{34}^{(p)} = 0.0, \lambda_{35}^{(p)} = 0.0, \\ \lambda_{36}^{(p)} &= 0.0, \lambda_{37}^{(p)} = 0.0, \lambda_{38}^{(p)} = 0.0, \lambda_{39}^{(p)} = 0.0, \lambda_{40}^{(p)} = 0.0, \\ \lambda_{41}^{(p)} &= 0.0 \end{aligned} \quad (1)$$

В формуле (1)  $\lambda_1^{(p)}, \dots, \lambda_{41}^{(p)}$  – это доли собственных чисел ковариационной матрицы, выраженные в процентах. Начиная с  $\lambda_{18}^{(p)}$  эта доля не превышает 0.2%. Это означает, что существует 24 компонента, которые вносят незначительный вклад в информативность данных. Метод *PCA* не представляет возможности точно определять какие именно признаки вносят существенный вклад в информативность данных, поэтому необходимо самостоятельно выбирать и удалять признаки с низким вкладом. Было сделано предположение, что наименее информативными признаками являются: `pre_pterm`, `pre_fterm`, `pre_loans_next_pay_summ`, `pre_loans_outstanding`, `pre_loans_total_overdue`, `pre_loans_max_overdue_sum`, `pre_loans_credit_cost_rate`, `is_zero_loans5`, `is_zero_loans530`, `is_zero_loans3060`, `is_zero_loans6090`, `is_zero_loans90`, `pre_util`, `pre_maxover2limit`, `is_zero_util`, `is_zero_over2limit`.

Для подтверждения необходимо повторно проделать метод *PCA* без 16 признаков, касательно которых было сделано предположение. В листинге 3 (см. приложение

- 1) реализуется метод главных компонент.
1. Строки 1–2. Задаётся файл с датасетом из 25 признаков, который считывается в объект `DataFrame(df)`.
2. Строки 4–10. Формируется список со статусами ежемесячных платежей `enc_paym_k`,  $k = 0, \dots, 24$ .
3. Страна 12. `X_pay = df.loc[:, columns_pay].copy()` – из исходного датасета `df` выбираются 25 признаков, которые формируют матрицу `X_pay`, содержащую информацию о платёжной дисциплине.
4. Страна 13. Создаётся объект метода главных компонент `PCA` с параметрами по умолчанию.
5. Страна 14. На матрице `X_pay` обучается модель `PCA`.
6. Страна 16. Вычисляются доли объясненной дисперсии для каждого главного компонента.

Получается следующая значимость признаков, процентно выраженная в собственных числах ковариационной матрицы `X_pay`.

$$\begin{aligned} \lambda_1^{(p)} &= 65.5, \lambda_2^{(p)} = 16.0, \lambda_3^{(p)} = 5.7, \lambda_4^{(p)} = 3.0, \lambda_5^{(p)} = 1.9, \\ \lambda_6^{(p)} &= 1.3, \lambda_7^{(p)} = 1.0, \lambda_8^{(p)} = 0.8, \lambda_9^{(p)} = 0.7, \lambda_{10}^{(p)} = 0.6, \\ \lambda_{11}^{(p)} &= 0.5, \lambda_{12}^{(p)} = 0.4, \lambda_{13}^{(p)} = 0.4, \lambda_{14}^{(p)} = 0.3, \lambda_{15}^{(p)} = 0.3, \\ \lambda_{16}^{(p)} &= 0.3, \lambda_{17}^{(p)} = 0.3, \lambda_{18}^{(p)} = 0.2, \lambda_{19}^{(p)} = 0.2, \lambda_{20}^{(p)} = 0.2, \\ \lambda_{21}^{(p)} &= 0.1, \lambda_{22}^{(p)} = 0.1, \lambda_{23}^{(p)} = 0.1, \lambda_{24}^{(p)} = 0.1, \lambda_{25}^{(p)} = 0.1. \end{aligned} \quad (2)$$

Из формулы (2) можно заметить, что осталось только 25 признаков, где доли собственных чисел  $\lambda_i^{(p)}$  не равны нулю, т.е. остались те признаки, которые вносят существенный вклад в информативность данных. Однако из таблицы 3 видно, что некоторые метрики для алгоритма случайного леса равны нулю, что является неприемлемым для прогнозирования кредитных рисков. Например, это метрики *Recall* и *Precision*, для дефолтных клиентов. На первый взгляд метод *PCA* не оказал влияния на некоторые метрики алгоритма, поэтому требуется дальнейший детальный анализ.

### 1.1.2 Определение аномального клиента

Рисунок 1 показывает, что доля недефолтных клиентов существенно велика и не соответствует статистике в реальной жизни. По данным Первого кредитного бюро около 20% <sup>2</sup> казахстанских заемщиков имеют дефолтную кредитную историю, что

---

<sup>2</sup>Можно ли "отбелить" кредитную историю и есть ли черный список должников в Казахстане [Электронный ресурс] / Информационный интернет-портал. URL: <https://tengrinesnews.kz/kazakhstan-news/li-otbelit-kreditnyu-istoriyu-est-chernyyiy-spisok-doljnikov-560869/> (дата обращения 25.11.2025).

подтверждает данное утверждение и указывает на наличие аномальности в классе недефолтных клиентов.

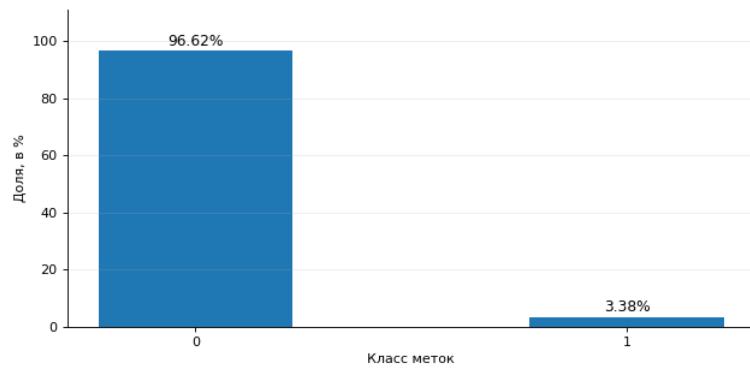


Рисунок 1 – Перекос в сторону недефолтных клиентов

Источник: составлено автором на основе: Соревнование на данных кредитных историй [Электронный ресурс] / Open Data Science. – URL: <https://ods.ai/competitions/dl-fintech-bki> (дата обращения: 25.11.2025).

В данной работе под аномальным клиентом понимается нетипичное поведение для большинства заемщиков. Речь идет о ситуации, когда клиент фактически не располагает достаточными денежными средствами для обслуживания кредита, однако продолжает вносить ежемесячные платежи с небольшими, но регулярными опозданиями, перезанимая необходимые суммы из других источников. В таблице 2 приведен пример неестественных выплат клиента, из которого видно, что платежная дисциплина клиента по отдельному кредиту не выглядит дефолтной, однако в среднем этот клиент не выплачивал обязательства пунктуально ни в один расчетный месяц (средние значения по месяцам представлены в последней строке таблицы). Ниже будет дано определение и алгоритм выявления таких клиентов в датасете.

**Определение 1.** Аномальным клиентом является такой клиент, для которого выполняются два условия:

$$\begin{cases} (a) \text{метка (флаг)} = 0, \\ (b) \min(\text{enc\_paym\_n}) > 0, \quad n = 0, \dots, 24. \end{cases} \quad (\text{anom})$$

В формуле (anom) условие (a) означает, что аномальный клиент в датасете является недефолтным, а условие (b) – по всем месяцам оплата кредитных обязательств проводилась с опозданием. В листинге 4 (см. приложение 1) указана реализация формулы *anom*.

**Комментарий 3.** Пояснение к листингу 4:

1. Стока 1. Читаем файл `train_target.csv` с целевой переменной в датафрейме `df_y`.

2. Строки 2–3. Из столбца `flag` датафрейма `df_y` извлекаются метки для первых 1000000 клиентов, где `y` – это исходные метки дефолта/недефолта, а `y_` – новые метки, которые будут изменяться в процессе нахождения аномальных клиентов.
3. Стока 5. Инициализируются счетчики `counter_norm` – число не аномальных клиентов, а `counter_anom` – число аномальных клиентов.
4. Стока 6. Создаются пустые списки, в которые будут записываться индексы не аномальных (`ind_norm`) и аномальных клиентов (`ind_anom`).
5. Стока 7. Запускается цикл по всем строкам матрицы `X_pay`, где `i` это порядковый номер клиента (индекс строки), а `x_pay` – массив платежной истории по всем месяцам.
6. Строки 8–11. Проверяется условие, которое определяет изначально недефолтных клиентов (`y_[i] == 0`), у которых по всем месяцам наблюдаются задержки платежей (`min(x_pay) > 0`). Если оба условия для клиента выполняются, то счетчик аномальных клиентов увеличивается на 1 (`counter_anom +=1`) и пополняется список индексов аномальных клиентов (`ind_anom.append(i)`), а его новой метке `y_[i]` присваивается значение 1. Таким образом, часть клиентов, имевших исходную метку 0, переводится в класс аномальных клиентов с меткой 1.
7. Строки 12–14. В случае неудовлетворения этим двум условиям цикл относит клиента к не аномальным (`counter_norm += 1`) и добавляет его индекс в список не аномальных клиентов (`ind_norm`).

Данный алгоритм показал, что доля дефолтных клиентов увеличилась с 3.38% до 35.45% (см. Рисунок 2). Это означает, что часть недефолтных клиентов перешла в класс дефолтных, и именно они удовлетворяют определению (*anom*).

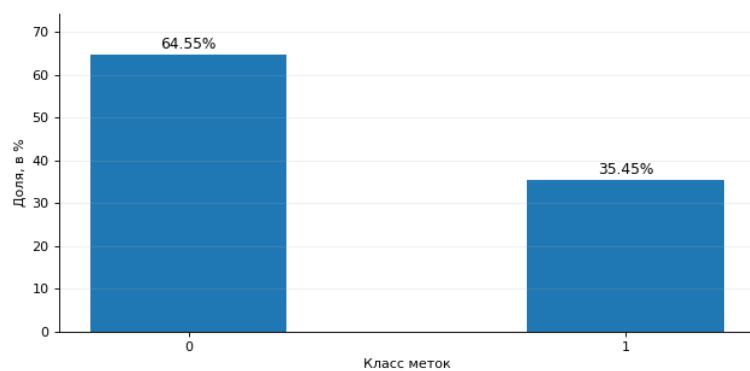


Рисунок 2 – Переход клиентов из класса «0» в класс «1»

Источник: составлено автором на основе: Соревнование на данных кредитных историй [Электронный ресурс] / Open Data Science. – URL: <https://ods.ai/competitions/dl-fintech-bki> (дата обращения: 25.11.2025).

Для большей наглядности результат переклассификации представлен на диаграмме Эйлера-Венна (см. Рисунок 3), показывающий пересечение множеств клиентов с дефолтными и недефолтными метками. Красный круг соответствует всем клиентам с исходной недефолтной меткой (96.62%), зеленый круг – клиентам с дефолтной меткой (3.38%), а их пересечение отражает группу аномальных клиентов (32.06%).

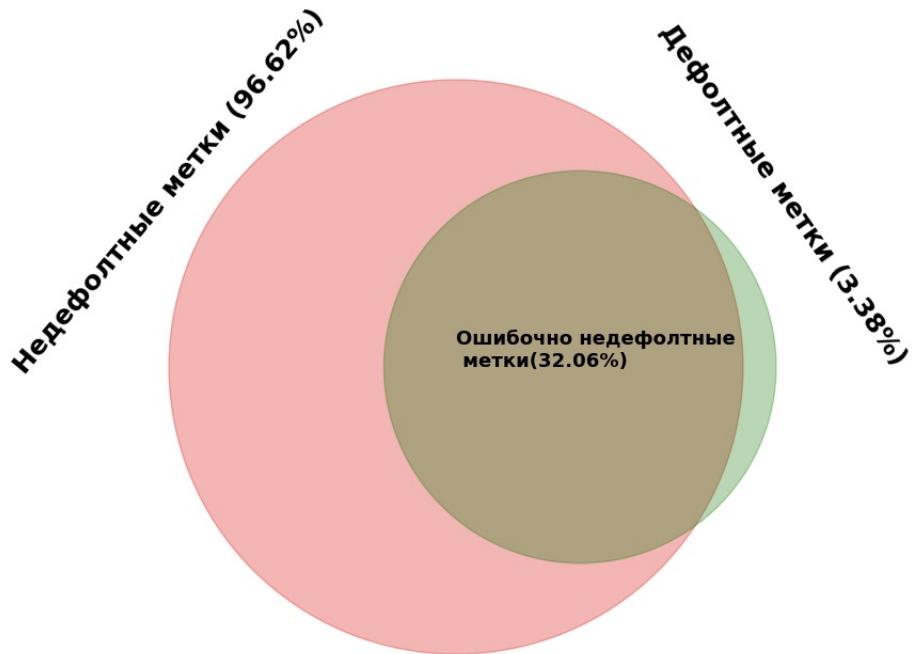


Рисунок 3 – Пересечение дефолтных и недефолтных меток с выделением ошибочно недефолтных клиентов

Источник: составлено автором на основе: Соревнование на данных кредитных историй [Электронный ресурс] / Open Data Science. – URL: <https://ods.ai/competitions/dl-fintech-bki> (дата обращения: 25.11.2025).

Далее необходимо посмотреть как отреагирует алгоритм случайного леса на данных преобразованиях согласно определению (*anom*). Под реакцией алгоритма понимается изменение точности метрик. После выявления аномальных клиентов качество модели значительно улучшилось (см. Таблицу 4).

Таблица 4 – Изменение точности метрик

Алгоритм – Случайный лес						
Flag метка	Precision точность	Recall полнота	f1-score f1-мера	Accuracy точность	ROC AUC	Количество
Тренировочная выборка						
0	0.99	0.97	0.98	0.97	0.97	483921
1	0.94	0.99	0.96			266079
Тестовая выборка						

0	0.99	0.97	0.98	0.97	0.97	161593
1	0.94	0.99	0.96			88407

Источник: составлено автором на основе: Соревнование на данных кредитных историй [Электронный ресурс] / Open Data Science. – URL: <https://ods.ai/competitions/dl-fintech-bki> (дата обращения: 25.11.2025).

Если сравнить первоначальные результаты, то можно заметить, что если до этого модель игнорировала класс дефолтных клиентов (все метрики равнялись нулю, см. Таблицу 3), то после переклассификации метрики точности стали высокими: на обучающей выборке  $precision = 0.94$ ,  $recall = 0.99$ ,  $f1 - score = 0.96$ , аналогичные значения показывает тестовая выборка. При этом можно качество недефолтного класса практически не ухудшилось:  $precision$  снизился с 1.00 до 0.99, а метрики  $recall$  и  $f1 - score$  не изменились. Также  $ROC AUC$  увеличился с 0.51 до 0.97, что свидетельствует о том, что теперь модель хорошо различает дефолтных и недефолтных клиентов. Одновременно увеличилось количество дефолтных клиентов в выборках за счет вновь выявленных аномальных наблюдений, что позволило алгоритму случайного леса показать эффективные результаты на обоих классах.

### 1.1.3 Статистический тест Колмогорова-Смирнова

Понятие аномального клиента, изложенное в разделе 1.1.2, основано на анализе платежного поведения заемщиков. Применение данного подхода привело к существенному улучшению метрик точности алгоритма случайного леса. Поэтому на следующем шаге проводится статистический анализ, позволяющий увидеть и обосновать, насколько поведение аномальных клиентов отличается от поведения дефолтных клиентов. Для проверки равенства (неравенства) функций распределения двух выборок применяется критерий Колмогорова-Смирнова. А.Н. Колмогоровым<sup>3</sup> и Н.В. Смирновым<sup>4</sup> была доказана следующая теорема:

**Теорема 1.** Пусть  $F_{X_1}(x)$ ,  $F_{X_2}(x)$  – выборочные функции распределения двух независимых выборок объемами  $n$  и  $m$ . Обозначим:

$$D_{n,m} = \sup_x |F_{X_1}(x) - F_{X_2}(x)|.$$

Тогда для любого  $t > 0$  выполняется:

$$\forall t > 0 : \lim_{n,m \rightarrow \infty} P\left(\sqrt{\frac{nm}{n+m}} D_{n,m} \leq t\right) = K(t) = \sum_{j=-\infty}^{+\infty} (-1)^j e^{-2j^2 t^2}. \quad (3)$$

<sup>3</sup>Колмогоров А.Н. (1903-1987) – Герой Социалистического Труда, профессор Московского государственного университета, академик АН СССР – крупнейший математик XX века, является одним из основоположников современной теории вероятности.

<sup>4</sup>Смирнов Н.В. (1900-1966) – член-корреспондент АН СССР, один из создателей непараметрических методов математической статистики и теории предельных распределений порядковых статистик.

На основе статистики  $D_{n,m}$  строится статистика критерия

$$t_{n,m} = \sqrt{\frac{nm}{n+m}} D_{n,m}.$$

С помощью этой статистики проверяются нулевая и альтернативная гипотезы о совпадении распределений двух выборок:

$$\begin{cases} \mathcal{H}_0 : F_{X_1}(x) = F_{X_2}(x), \\ \mathcal{H}_1 : F_{X_1}(x) \neq F_{X_2}(x). \end{cases}$$

При заданном уровне значимости  $\alpha$  выбирается критическое значение  $K_\alpha$  распределения Колмогорова. Правило принятия решения имеет следующий вид:

$$\begin{cases} t_{n,m} \leq K_\alpha, & \text{нулевая гипотеза } \mathcal{H}_0 \text{ принимается,} \\ t_{n,m} > K_\alpha, & \text{нулевая гипотеза } \mathcal{H}_0 \text{ отвергается в пользу } \mathcal{H}_1. \end{cases}$$

Для применения критерия Колмогорова-Смирнова клиенты были разделены на две подвыборки:

$$\begin{cases} X_1 = X_{1 \rightarrow 1}, & (\text{default}), \\ X_2 = X_{0 \rightarrow 1}, & (\text{anom}), \end{cases}$$

где  $X_1$  – выборка клиентов, являющихся дефолтными до и после условия аномальности;  $X_2$  – выборка клиентов, у которых метка изменилась с «0» на «1».

Задача состоит в том, чтобы выяснить, принадлежат ли выборки  $X_1$  и  $X_2$  одному распределению, применив критерий Колмогорова-Смирнова. Если выборки  $X_1$  и  $X_2$  принадлежат одному распределению (принятие гипотезы  $\mathcal{H}_0$ ), то платежная дисциплина дефолтных и аномальных клиентов одинакова. Критерий Колмогорова-Смирнова реализуется в листинге 5 (см. приложение 1).

Датасет содержит 25 дисциплинарных признаков ( $enc\_paym\_0, \dots, enc\_paym\_24$ ), и для каждого из них необходимо применить данный тест, чтобы выяснить, для каких признаков принимается та или иная гипотеза. Поскольку в датасете 3 000 000 наблюдений (объемы выборок  $n$  и  $m$  велики), то при уровне значимости  $\alpha = 0.05$  в качестве критического значения используется  $K_\alpha = 1.36$ .

Анализ показал, что первые 15 месяцев ( $enc\_paym\_0, \dots, enc\_paym\_14$ ) поведение аномальных клиентов отличается от поведения дефолтных, а начиная с 16 месяца ( $enc\_paym\_15, \dots, enc\_paym\_24$ ), поведения становятся схожими. Так, например, для 15 месяца ( $enc\_paym\_14$ ) (см. Рисунок 4) расчетное значение статистики  $t_{n,m} = 1.41$ . Поскольку  $K_\alpha = 1.36$  и  $t_{n,m} > K_\alpha$ , принимается гипотеза  $\mathcal{H}_1$ . Это означает, что финансовое поведение аномальных клиентов, ранее относившихся к классу «0», отличается

от поведения дефолтных клиентов.

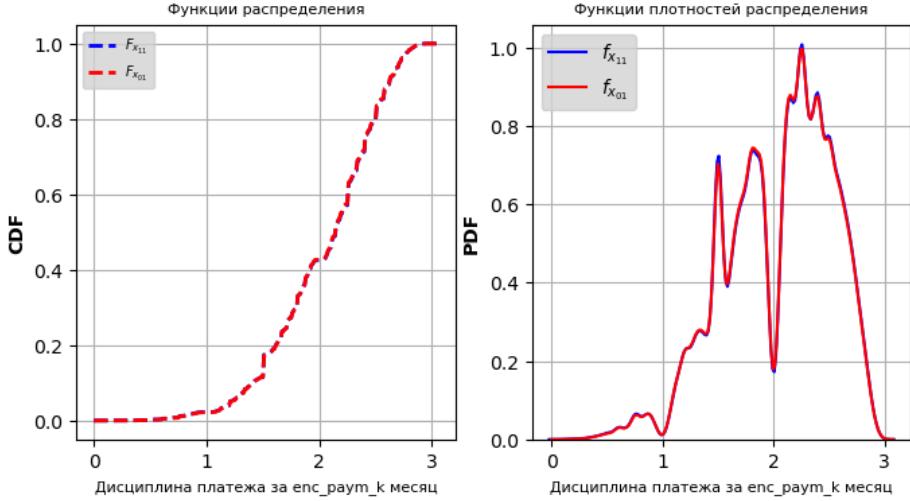


Рисунок 4 – Функции распределения и функции плотности за enc\_paym\_14 месяц

Источник: составлено автором на основе: Соревнование на данных кредитных историй [Электронный ресурс] / Open Data Science. – URL: <https://ods.ai/competitions/dl-fintech-bki> (дата обращения: 25.11.2025).

Для 16 месяца ( $enc\_paym\_15$ ) (см. Рисунок 5) значение  $t_{n,m} = 1.31$ . В этом случае  $t_{n,m} < K_\alpha$ , поэтому принимается гипотеза  $\mathcal{H}_0$ . Распределения значений признака для аномальных и дефолтных клиентов не различаются.

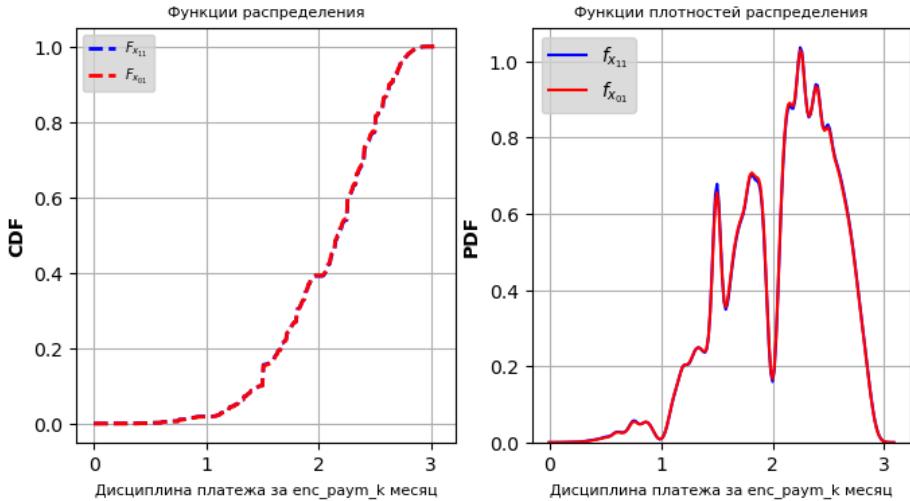


Рисунок 5 – Функции распределения и функции плотности за enc\_paym\_15 месяц

Источник: составлено автором на основе: Соревнование на данных кредитных историй [Электронный ресурс] / Open Data Science. – URL: <https://ods.ai/competitions/dl-fintech-bki> (дата обращения: 25.11.2025).

Таким образом, выплаты аномальных и дефолтных клиентов до 15 месяца не соответствуют друг другу, а начиная с 16 месяца их поведение становится схожим.

## 1.2 Базовые алгоритмы оценки вероятности возврата кредита

1.2.1 Деревья решений

1.2.2 Случайный лес

1.2.3 Градиентный бустинг

1.2.4 Нейронные сети

## СПИСОК ЛИТЕРАТУРЫ

1. *Давыдова А. А.* Анализ и оценка кредитоспособности заемщика финансовыми организациями: обзор подходов и методов оценки [Электронный ресурс] // Вестник ВУ-иТ. — 2023. — Т. 51, № 1. — URL: <https://cyberleninka.ru/article/n/analiz-i-otsenka-kreditosposobnosti-zaemschika-finansovymi-organizatsiyami-obzor-podhodov-i-metodov-otsenki>. — Загл. с экрана.
2. Можно ли «отбелить» кредитную историю и есть ли черный список должников в Казахстане [Электронный ресурс] / Информационный интернет-портал. — URL: [https://tengrnews.kz/kazakhstan\\_news/li-otbelit-kreditnuyu-istoriyu-est-chernyyiy-spisok-doljnikov-560869/](https://tengrnews.kz/kazakhstan_news/li-otbelit-kreditnuyu-istoriyu-est-chernyyiy-spisok-doljnikov-560869/). — Загл. с экрана.

## Приложение 1. Предобработка данных

Листинг 1 – Формирование единого датасета

```
1 import pandas as pd
2 import os
3 import pyarrow.parquet as pq
4
5 path = "train_data"
6 for i,file in enumerate(os.listdir(path)):
7     file = 'train_data_' + str(i) + '.pq'
8     dataset = pq.ParquetDataset(os.path.join(path,file))
9     df = dataset.read(use_threads=True).to_pandas()
10    df_gr = df.groupby('id').agg('mean')
11    file_csv = file.replace('.pq', '.csv')
12    df_gr.to_csv(os.path.join('train_data_csv_all',file_csv))
13 os.listdir(path)
14
15 path = 'train_data_csv_all'
16 frames = []
17 for file_csv in os.listdir(path):
18     if file_csv.endswith('.csv'):
19         df = pd.read_csv(os.path.join(path, file_csv))
20         frames.append(df)
21
22 result = pd.concat(frames)
23 file_csv_all = '1_data_csv_all.csv'
24 result.to_csv(file_csv_all)
25 df_all = pd.read_csv(file_csv_all)
```

Листинг 2 – Датасет, содержащий 41 признак

```
1 from sklearn.decomposition import PCA
2
3 file_csv = 'data_csv_small.csv'
4 df = pd.read_csv(file_csv)
5
6 columns = ['pre_pterm', 'pre_fterm',
7             'pre_loans_next_pay_summ', 'pre_loans_outstanding',
8             'pre_loans_total_overdue', 'pre_loans_max_overdue_sum',
9             'pre_loans_credit_cost_rate', 'is_zero_loans5',
10            'is_zero_loans530', 'is_zero_loans3060',
11            'is_zero_loans6090', 'is_zero_loans90',
12            'pre_util', 'pre_maxover2limit',
13            'is_zero_util', 'is_zero_over2limit',
14            'enc_paym_0', 'enc_paym_1', 'enc_paym_2',
15            'enc_paym_3', 'enc_paym_4', 'enc_paym_5',
16            'enc_paym_6', 'enc_paym_7', 'enc_paym_8',
17            'enc_paym_9', 'enc_paym_10', 'enc_paym_11',
18            'enc_paym_12', 'enc_paym_13', 'enc_paym_14',
19            'enc_paym_15', 'enc_paym_16', 'enc_paym_17',
20            'enc_paym_18', 'enc_paym_19', 'enc_paym_20',
21            'enc_paym_21', 'enc_paym_22',
22            'enc_paym_23', 'enc_paym_24']
23
24 X = df.loc[:, columns].copy()
25 pca = PCA()
26 pca.fit(X)
27
28 100*pca.explained_variance_ratio_.round(3)
```

Листинг 3 – Датасет, содержащий 24 признака

```
1 file_csv = 'data_csv_small.csv'
2 df = pd.read_csv(file_csv)
3
4 columns_pay = ['enc_paym_0', 'enc_paym_1', 'enc_paym_2',
5                 'enc_paym_3', 'enc_paym_4', 'enc_paym_5',
6                 'enc_paym_6', 'enc_paym_7', 'enc_paym_8',
7                 'enc_paym_9', 'enc_paym_10', 'enc_paym_11',
8                 'enc_paym_12', 'enc_paym_13', 'enc_paym_14',
9                 'enc_paym_15', 'enc_paym_16', 'enc_paym_17',
10                'enc_paym_18', 'enc_paym_19', 'enc_paym_20',
11                'enc_paym_21', 'enc_paym_22', 'enc_paym_23',
12                'enc_paym_24']
13
14 X_pay = df.loc[:, columns_pay].copy()
15 pca = PCA()
16 pca.fit(X_pay)
17
18 100*pca.explained_variance_ratio_.round(3)
```

Листинг 4 – Алгоритм выявления аномальных клиентов

```
1 df_y = pd.read_csv('train_target.csv')
2 y = df_y.flag.values[:1000000].copy()
3 y_ = df_y.flag.values[:1000000].copy()
4
5 counter_norm, counter_anom = 0, 0
6 ind_norm, ind_anom = [], []
7 for i,x_pay in enumerate(np.array(X_pay)):
8     if y_[i]==0 and min(x_pay)>0:
9         counter_anom +=1
10        ind_anom.append(i)
11        y_[i]=1
12    else:
13        counter_norm +=1
14        ind_norm.append(i)
```

Листинг 5 – Критерий Колмогорова-Смирнова

```
1 col = 'enc_paym_15'
2 T = X_pay.enc_paym_15
3 x00, x01 = T[(y==0) & (y_== 0)], T[ind_anom]
4 x11, x01 = T[(y==1)], T[ind_anom]
5 ind00 = (x00!=0) & (x00!=1) & (x00!=2) & (x00!=3) & (x00!=4) & (x00
   !=5)
6 ind01 = (x01!=0) & (x01!=1) & (x01!=2) & (x01!=3) & (x01!=4) & (x01
   !=5)
7 ind11 = (x11!=0) & (x11!=1) & (x11!=2) & (x11!=3) & (x11!=4) & (x11
   !=5)
8 x00, x11, x01 = x00[ind00], x11[ind11], x01[ind01]
9
10 def my_cdf(y11, y211, step):
11     tm = max( max(y11), max(y211) )
12     F1, F2 = [], []
13     for t in np.arange(0, tm+0.1, step):
14         F1.append( y11[(y11>0)&(y11<t)].shape[0] )
15         F2.append( y211[(y211>0)&(y211<t)].shape[0] )
16     F1, F2 = np.array(F1)/y11[(y11>0)].shape[0], np.array(F2)/y211[(
       y211>0)].shape[0]
17     n1, n2 = y11[y11>0].shape[0], y211[y211>0].shape[0]
18     K = max(abs(F1-F2))*np.sqrt( n1*n2/(n1+n2) )
19     return F1, F2, K, tm
20 step = 0.01
21 F01, F11, K, tm = my_cdf( x01, x11, step )
22 print(K)
```

Приложение 2