|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

|  |  |
| --- | --- |
|  | «Утверждено»  Правлением  ОАО «Банк Эсхата»  Протокол \_\_\_\_от\_\_\_\_\_\_\_2024г. |
|

**РУКОВОДСТВО Р-000:2024**

по разработки кредитного скоринга

для розничного сегмента (Версия 01)

«Самозанятые»

**СОДЕРЖАНИЕ**

[1. ОБЩИЕ ПОЛОЖЕНИЯ 2](#_Toc162355406)

[2. ОБЗОР СОДЕРЖАНИЯ ДОКУМЕНТА 4](#_Toc162355407)

[3. МЕТОДОЛОГИЯ ПОСТРОЕНИЯ МОДЕЛЕЙ КРЕДИТНОГО СКОРИНГА 5](#_Toc162355408)

[4. ПРИЛОЖЕНИИ 6](#_Toc162355409)

[ПРИЛОЖЕНИЕ №1. ОБЗОР МЕТОДОЛОГИИ КРЕДИТНОГО СКОРИНГА ДЛЯ СТАТИСТИЧЕСКОЙ МОДЕЛИ 7](#_Toc162355410)

[ПРИЛОЖЕНИЕ №2. ПОСТАНОВКА ЗАДАЧИ 9](#_Toc162355411)

[ПРИЛОЖЕНИЕ №3. ОПИСАНИЕ И ИМПОРТ ДАННЫХ 10](#_Toc162355412)

[ПРИЛОЖЕНИЕ №4. ПРЕ-ПРОЦЕССИНГ ДАННЫХ 12](#_Toc162355413)

[ПРИЛОЖЕНИЕ №5. МОДЕЛИРОВАНИЕ. ДЛЯ «САМОЗАНЯТЫЕ» 22](#_Toc162355414)

[ПРИЛОЖЕНИЕ №6. СКОРИНГОВАЯ КАРТА И КАЛИБРАЦИЯ. «САМОЗАНЯТЫЕ» 36](#_Toc162355415)

[ПРИЛОЖЕНИЕ №7. БИЗНЕС-СТРАТЕГИИ 40](#_Toc162355416)

[ПРИЛОЖЕНИЕ №8. ВЫВОДЫ 42](#_Toc162355417)

# 

# ОБЩИЕ ПОЛОЖЕНИЯ

**1.1. Определения**

**DPD N+ Ever**- просрочка по кредиту более N дней за любой период на балансе.

**Exploratory Data Analysis** – это анализ основных свойств данных, нахождение в них общих закономерностей, распределений и аномалий, построение начальных моделей, зачастую с использованием инструментов визуализации

**PD (Probability of Default)** - вероятность дефолта

**Баллы отсечения** — это определенные значения, которые используются для принятия решения о выдаче кредита или других финансовых услуг на основе результатов скоринговой модели. Эти баллы определяют, является ли заемщик кредитоспособным или нет.

**Бинарная классификация** — это задача классификации элементов заданного множества в две группы (предсказание, какой из групп принадлежит каждый элемент множества) на основе правила классификации.

**Биннинг** — это процесс категоризации, суть которого заключается в разбиении переменных определенным образом на группы.

**Бэк-тестинг** — это процесс оценки производительности скоринговой модели на исторических данных. Это позволяет оценить, насколько хорошо модель предсказывает результаты на прошлых данных, что может помочь в оптимизации модели для будущих прогнозов.

**Валидация** – это проверка правильности работы (предсказательной способности) аналитической модели, построенной на основе машинного обучения, а также удостоверение, что она соответствует требованиям решаемой задачи

**Выборка** — это процесс выбора определенного набора данных из общего объема информации для анализа или исследования.

**Выбросы** — это значения в данных, которые существенно отличаются от остальных и могут искажать статистические показатели.

**Гетероскедастичность** – это понятие, используемое в прикладной статистике (чаще всего — в эконометрике), означающее неоднородность наблюдений, выражающуюся в неодинаковой (непостоянной) дисперсии случайной ошибки регрессионной (эконометрической) модели.

**Дефолт** - Понятие определенное, согласно кредитной политике.

**Зашумленные данные** — это данные, которые содержат некоторый уровень случайного или нежелательного шума, что может затруднить их анализ или использование. Шум может возникать из-за ошибок измерения, внешних воздействий, или других факторов.

**Итерация** — это повторение какого-либо действия. Итерация в математике — повторное применение какой-либо математической операции

**Калибрация** — это процесс настройки или коррекции измерительных приборов, чтобы они показывали точные результаты.

**Калибровка** — это процесс корректировки скоринговых моделей для улучшения их точности и надежности.

**Логистическая регрессия** — это статистический метод для анализа данных, который используется для прогнозирования вероятности возникновения определенного события на основе одного или нескольких независимых переменных.

**Предиктор (feature engineering)** — это прогностический параметр; средство прогнозирования.

**Пре-процессинг (Data Engineering)** — это процесс подготовки и очистки данных перед их анализом или использованием в машинном обучении.

**Семплинг** — это процесс взятия образцов или выборки из определенного набора данных для последующего анализа или использования.

**Скрипт** — это понятие в программировании, обозначающее последовательность команд для выполнения конкретных операций.

**Целевая переменная** — это переменная, которую мы пытаемся предсказать или моделировать в статистике, машинном обучении или других областях анализа данных.

**Язык R** – это язык программирования с открытым исходным кодом, который широко используется как статистическое программное обеспечение и инструмент анализа данных.

# ОБЗОР СОДЕРЖАНИЯ ДОКУМЕНТА

Данный документ разработан в рамках проекта по разработке системы оценки кредитного риска — кредитного скоринга — для продуктов розничного сегмента Банка. Результатом работы системы кредитного скоринга является расчет оценки вероятности дефолта (PD) на этапе подачи потенциальным заемщиком заявки на рассмотрение кредита.

Текущий документ содержит:

* описание общего процесса разработки системы кредитного скоринга на основе классического подхода в виде скоринговой карты
* описание построенной модели от пре-процессинга данных до валидации модели и получения скоринговых баллов
* разбор этапов построения скоринговой модели на языке R с элементами скрипта и комментариям к нему
* описание принципа принятия решений на этапе построения модели.

Данный документ является руководством пользователя на примере конкретной выборки данных, предикторов и целевых переменных. Все выводы релевантный к предложенной выборке. Общий процесс и этапы скрипта могут применяться к любой задаче бинарной классификации (целевая переменная имеет логику 0/1, Да/Нет).

Дата выгрузки данных — 30.06.2023г

Дата первой итерации скоринговых моделей — 25.09.2023г

Дата финализации скоринговых моделей — 13.10.2023г

Дата финальной документации —01.04.2024г

# МЕТОДОЛОГИЯ ПОСТРОЕНИЯ МОДЕЛЕЙ КРЕДИТНОГО СКОРИНГА

Процесс построения скоринговой модели описан в следующей схеме:



**3.1. Постановка бизнес-задачи.**

Какой показатель, метрика прогнозируется? Для каких целей и как планируется использоваться? Для какого сегмента клиентов, заявок, продуктов?

**3.2. Анализ первичных данных.**

Какие данные мы имеем? В каких форматах и где они хранятся? Могут ли они быть собраны и агрегированы на уровне клиента, заявки, договора?

**3.3. Настройка среды разработки. Импорт данных.**

а) Установка среды разработки (Rstudio);

b) Настройка и установка необходимых библиотек;

c) Импорт данных из Excel- а в Rstudio;

**3.4. Подготовка и пре-процессинг данных (****Data Engineering)**

a) Подготовка дата семпла;

b) Пре-процессинг выборки данных;

c) Объединение данных из различных источников;

d) Удаление дублей;

e) Обработка зашумленных данных;

f) Обработка пропущенных значений, выбросов;

g) Расчет предикторов (feature engineering);\*

h) Создание производных переменных из оригинальных (кросс-переменные, логические переменные);

i) Выбор целевой переменной (target variable);

j) Определение бинарной переменной (0/1) исходя из критерия события — дефолта по кредиту

k) Разделение на обучающую и тестовую выборку (train and test samples);

l) Разделение выборки данных на под выборки для обучения и тестирования

**3.5. Моделирование**

a) Биннинг переменных (binning: fine classing and coarse classing) \*

Приведение всех переменных к виду категориальных переменных

b) Выбор переменных для модели \*

Отбор значимых переменных уменьшения объема выборки

c)Обучение и выбор модели классификации (model training and selection)

Запуск алгоритмов поиска параметров моделей бинарной классификации

d) Валидация модели

Расчет метрик прогнозной силы, точности, и стабильности модели

**3.6. Скоринговая карта (scorecard)**

a) Приведение модели к формату скоринговой карты \*

Классический подход к кредитному скорингу на основе линейной модели логистической регрессии

b) Калибрация скоринговой модели \*

Приведение скоринговых баллов модели к выбранной шкале. Корректировка оценки прогноза вероятности события с помощью линейного преобразования

**3.7. Внедрение**

a) Техническое внедрение модели (deployment)

Выбор архитектуры для запуска модели в промышленную эксплуатацию

b) Внедрение бизнес-стратегии (use strategies, business implementation)

Построение распределений потока заявок. Определение параметров управлений (контролей) для принятия решений: баллы отсечения, сегменты.

**3.8. Мониторинг модели**

Наблюдение и диагностика стабильности модели и популяции, качества работы модели, стратегий.

Этапы обозначенные \* не являются обязательными.

Вышеприведенный процессы детально раскрыты в Приложениях к данному документу.

## 4. ПРИЛОЖЕНИИ

## ПРИЛОЖЕНИЕ №1. ОБЗОР МЕТОДОЛОГИИ КРЕДИТНОГО СКОРИНГА ДЛЯ СТАТИСТИЧЕСКОЙ МОДЕЛИ

Задачей кредитного скоринга является предсказание (оценка) вероятности дефолта — события неплатежа заемщиком по кредитном договору. Кредитный скоринг является задачей бинарной классификации. Для прогноза вероятности события используются предикторы — параметры заявки и кредитной истории клиента. Модель строится на исторический выборке предыдущих выдач, погашений, и дефолтов.

Для разработки системы скоринга создается выборка данных характеристик (или предикторов) и целевая переменная (индикатор «хорошо»/ «плохо»). Мы использует подход, ставший традиционным для задач кредитного скоринга — использование биннинга переменных (трансформация в категориальные переменные) и логистической регрессии для получения интуитивно понятного и объясняющего инструмента.

|  |  |
| --- | --- |
|  | Функции плотности популяции или частотные распределения портфеля по характеристикам группируются для получения более стабильных сегментов.  Вычисляется мера события для определенного параметра (Weight of Evidence). Это будет прогностическая мера мощности (информационная ценность) для дальнейшего включения в модель. |
| **Рисунок №1. Графический пример биннинга**  http://upload.wikimedia.org/wikipedia/commons/a/ac/Logistic-curve.png | Логистическая регрессия применяется для прогнозирования вероятности события.  Значения этой модели находятся в диапазоне от 0 до 1.  Функции распределения считаются приближенными к нормальному распределению. |
| **Рисунок №2. Графический пример логистической регрессии** | Необходимо найти w – веса регрессии для всех факторов x.  Вероятность события p используется для прогнозирования потерь и стратегий. |
| |  |  | | --- | --- | | **Возраст** | | | 18 - 23 | 9 | | 24 - 27 | 17 | | 28 - 34 | 23 | | 35 - 44 | 29 | | 45 - 59 | 37 | | 60 - больше | 39 | | **Пол** | | | M | 7 | | Ж | 12 | | **Доход** | | | 0–1000 | 5 | | 1001–2000 | 12 | | 2001 - 5000 | 19 | | 5001 - подробнее | 25 | | После оценки и калибровки параметров логистической регрессии модель может быть представлена в виде системы показателей, которая уже стала отраслевым стандартом.  Общий балл рассчитывается как общее количество всех весов атрибутов. |
| **Рисунок №3. Пример скоринговой карты** |  |
|  | Прогностическая сила скоринговой модели может быть оценена с использованием различных статистических коэффициентов и теста.  Наиболее популярными являются:  Тест Холмогорова-Смирнова – максимальная разница между кумулятивным распределением «плохих» и «хороших» и Коэффициент Джини – кумулятивное соотношение плохих и хороших кредитов. |
| **Рисунок №4. Графический пример теста**  **Холмогорова-Смирнова и коэффициента Джини**    **Рисунок №4. Графический пример теста Холмогорова-Смирнова и коэффициента Джини** | Кривая стратегии является определяющей для каждого определенного портфеля.  Это зависимость между коэффициентом принятия и кумулятивным коэффициентом уровня просроченной задолженности.  Пример формулы расчета вероятности дефолта для шкалы FICO. |

## ПРИЛОЖЕНИЕ №2. ПОСТАНОВКА ЗАДАЧИ

Перед Банком стоит задача увеличения объемов розничного кредитного портфеля при сохранении либо улучшении его качества. Кредитный портфель в нынешнем состоянии карты продукта и анкетных данных клиентов существует с октября 2021 года до даты начала проекта — июнь 2023 года. Качество кредитного портфеля высокое — показатель DPD 90+ Ever (просрочка по кредиту более 90 дней за любой период на балансе) колеблется на уровне 1%–2.3%. Улучшение текущей ситуации рассматривается в основном за счет повышения скорости обработки заявки.

Проведенная валидация используемой системы оценки кредитных рисков клиента показала существенное снижение качества скоринговой системы (см. Отчет о валидации системы кредитного скоринга, 2023). Модель аппликационного кредитного скоринга нуждается в обновлении.

Розничный продукт с точки зрения оценки рисков делится на три сегмента по источнику происхождения доходов клиента: работник организации, самозанятые, и получатели денежных переводов.

Задачей данного проекта является разработка системы аппликационного кредитного скоринга на основе имеющихся данных: анкетные данные клиентов, данные продукта, поведенческие данные (данные погашения кредитной задолженности).

Результатом работы модели кредитного скоринга является оценка вероятности дефолта заемщика на основе предоставленных на момент расчета оценки параметров. Событием дефолта является событие, определенное Банком как дефолт по кредиту, например, превышение количества дней просроченной задолженности по кредиту более, чем на 90 дней.

В качестве модели кредитного скоринга выбрана скоринговая карта. Данный подход является классическим решением в области кредитного скоринга.

Он позволяет:

1) внедрять модель оценки рисков в любой инфраструктуре (расчет возможен вручную или в таблицах Excel),

2) интерпретировать результаты оценки и анализировать рисковый профиль клиента,

3) получать стабильные оценки рискового профиля клиента в силу применения линейных моделей.

Скоринговая модель (карта) является инструментом оценки рисков, т. е. дает оценку вероятности события неплатежа по кредиту. Скоринговая модель не влияет напрямую на уровень просроченной задолженности (качество) портфеля. После внедрения скоринговой модели необходимо провести расчет и внедрение стратегии кредитования — определить сегменты принятия решений, установить баллы отсечений. Это производится на основании стратегической кривой, которая показывает соотношение уровня одобрения заявок и прогноз уровня просроченной задолженности по сегменту выдачи, и ее форма, кроме качества скоринговой модели, определяется текущим распределением популяции клиентов.

# ПРИЛОЖЕНИЕ №3. ОПИСАНИЕ И ИМПОРТ ДАННЫХ

Для разработки модели кредитного скоринга используется выборка данных из 3-х источниках:

* Данные заявки (параметры заявки и договора)
* Данные клиента (анкетные данные)
* Поведенческие данные погашения договора (в т. ч. уровень просрочки)

Всего 235,698 наблюдений по всему розничному портфелю за период 2021/10–2023/06.

Процесс анализа построен следующим образом:

1) весь набор данных по всем сегментам продуктов и источникам экспортируется в единую выборку на уровне договора,

2) проводится пре-процессинг — анализ качества, обработка переменных — на уровне всей выборки кредитного портфеля,

3) дальнейший процесс построения скоринговой модели: биннинг, выбор переменных, моделирование, валидация — проводится для каждого сегмента отдельно, и отражается в документации, соответствующей конкретному сегменту.

|  |  |
| --- | --- |
| **Имя переменной** | **Источник данных** |
| Customer ID | Account Data |
| Application ID | Account Data |
| Account ID | Account Data |
| Branch ID | Account Data |
| Product ID | Account Data |
| Application date | Account Data |
| Date loan granted | Account Data |
| Loan Amount | Account Data |
| First instalment due date | Account Data |
| Interest rate | Account Data |
| Collateral type | Account Data |
| Value of collateral | Account Data |
| Property type/Collateral type | Account Data |
| Salary payment in bank account | Account Data |
| Loan type | Account Data |
| # of instalments | Account Data |
| Instalment amount/Min instalment amount | Account Data |
| Customer ID | Application Data |
| Application ID | Application Data |
| Account ID | Application Data |
| Date of birth | Application Data |
| Gender | Application Data |
| City of Living | Application Data |
| Region of living | Application Data |
| City of registration | Application Data |
| Region of registration | Application Data |
| Work phone number | Application Data |
| Mobile phone number | Application Data |
| Education | Application Data |
| Marital status | Application Data |
| # Dependants | Application Data |
| # Children | Application Data |
| # months at current address | Application Data |
| Employment type | Application Data |
| Employment sector | Application Data |
| Employment segment | Application Data |
| # months at job | Application Data |
| Net main income | Application Data |
| Source of main income | Application Data |
| Additional income | Application Data |
| Source of additional income | Application Data |
| Reported expenses | Application Data |
| Months with bank | Application Data |
| Current exposure | Application Data |
| Client type | Application Data |
| Property object | Application Data |
| Eskhata Online flag | Application Data |
| Plastic Card flag | Application Data |
| Deposit flag | Application Data |
| Run Date | Behavioural Data |
| Customer ID | Behavioural Data |
| Account ID | Behavioural Data |
| Date account opened | Behavioural Data |
| Current balance | Behavioural Data |
| Date last payment | Behavioural Data |
| Date final payment | Behavioural Data |
| Due date | Behavioural Data |
| Payment amount | Behavioural Data |
| Account status | Behavioural Data |
| # of payments in arrears | Behavioural Data |
| Cumulative delinquency | Behavioural Data |
| Amount due – Instalment | Behavioural Data |
| Principal amount | Behavioural Data |
| Interest accrued | Behavioural Data |
| Outstanding balance | Behavioural Data |
| Arrears amount | Behavioural Data |
| Current days past due | Behavioural Data |
| Maximum days past due | Behavioural Data |
| Maximum days past due lifetime | Behavioural Data |
| Default flag | Behavioural Data |
| #prolongations | Behavioural Data |
| BKI Rating | Behavioural Data |
| BKI Number Loans | Behavioural Data |

Исходный набор данных содержит 74 поля, из которых 38 потенциальных предикторов.

Общее количество наблюдений — 235,698.

Период выборки — октябрь 2021 — июнь 2023. Дата выгрузки 30.06.2023.

# ПРИЛОЖЕНИЕ №4. ПРЕ-ПРОЦЕССИНГ ДАННЫХ

**1. Расчет предикторов (feature engineering)**

Исходя из имеющих данных, рассчитывается несколько потенциально значимых предикторов.

**Возраст** как разницу между датой рождения и датой заявки.

Библиотека lubridate содержит функцию time\_length для расчета разницы между датами в днях, месяцах, годах. В данном случае используем параметр "years".

library(lubridate)

df$Age = floor(time\_length(difftime(df$Application.date, df$Date.of.birth), "years" ))

Создается поле LoanMonth - Год и Месяц договора для удобства обработки по периодам. Например, 202105, 202106.

df$LoanMonth <- year(df$Application.date)\*100 + month(df$Application.date)

Создается переменные на основе других переменных.

Логическая переменная Равенство города проживания городу регистрации (CityOfLivingEqRegistration) - принимает значение 1, если город проживание совпадает с городом регистрации, и 0 в противном случае

df$CityOfLivingEqRegistration <- ifelse(as.character(df$City.of.Living) == as.character(df$City.of.registration), 1, 0)

Логическая переменная IsCollateral - принимает значение 0, если поле Collateral.type не заполнено (нет залога), в противном случае 1

df$IsCollateral <- ifelse(is.na(df$Collateral.type),0,1)

Создается переменные на основе поведения клиента по предыдущим договорам - Behavioural Features — Новый/Существующий клиент (New/Exist) та Максимальная предыдущая просрочка по договорам в банке (Previous MaxDPD)

# sqldf("select count(\*) from (select distinct [Customer.ID.x] from df) t1")

Создается рабочая таблица: связка договор - все предыдущие договора клиента

temp\_df <-

sqldf("select t1.[Account.ID] as [Account.ID],

t1.[Customer.ID.x] as [Customer.ID.x],

t1.[Application.date] as [Application.date],

t2.[Customer.ID.x] as CustID\_2,

t2.[Application.date] as AppDate\_2,

t2.[Maximum.days.past.due.lifetime]

from df t1

left join df t2

on t1.[Customer.ID.x] = t2.[Customer.ID.x] and t1.[Application.date] > t2.[Application.date]

order by t1.[Customer.ID.x]")

nrow(df)

nrow(temp\_df)

# View(temp\_df)

Рассчитывается переменная Максимальное количество дней в просрочке по предыдущим договорам (PreviousLoans\_MaxDPD) и количество предыдущих договоров (PreviousLoans\_Count).

Если текущий договор первый - возвращаем NULL.

Acc\_Numb\_before <-

sqldf("select t1.[Account.ID], count(\*) as cnt, max([Maximum.days.past.due.lifetime]) as Max\_MaxDaysPastDue

from temp\_df t1

where CustID\_2 is not null

group by t1.[Account.ID]

order by t1.[Account.ID]

")

Создается переменная "Существующий клиент". Если нет договоров до текущего договора - клиент новый, ExistingClient = 0

Acc\_Numb\_before$ExistingClient <- ifelse(Acc\_Numb\_before$cnt>1, 1, 0)

Присоединяется к основной таблице. Как и ранее - через служебную таблицу, чтобы не перезаписать

df\_new <-

sqldf("select t1.\*, t2.cnt as PreviousLoans\_Count, t2.Max\_MaxDaysPastDue as PreviousLoans\_MaxDPD

from df as t1 left join Acc\_Numb\_before as t2 on t1.[Account.ID] = t2.[Account.ID]")

Проверяется кол-во строк. Не должно увеличиться

nrow(df\_new)

Создается переменная кол-во предыдущих договоров

PreviousLoans\_Count

df\_new$PreviousLoans\_Count <- ifelse(is.na(df\_new$PreviousLoans\_Count), 0, df\_new$PreviousLoans\_Count)

Присваивается в df объединенную таблицу

df <- df\_new

**Целевые переменные**

Создается целевые переменные GB на основе счетчиков дней просрочки Maximum.days.past.due.lifetime и CumulativeDelinquency, и

# Если количество дней просрочки превышает пороговое значение – ставится 1 (плохой договор, событие), в противном случае — 0

df$GB\_90Ever <- ifelse(df$Maximum.days.past.due.lifetime >= 90, 1, 0)

df$GB\_CumDlq90 <- ifelse(df$CumulativeDelinquency >= 90, 1, 0)

df$GB\_60Ever <- ifelse(df$Maximum.days.past.due.lifetime >= 60, 1, 0)

**2. Exploratory Data Analysis и очистка данных**

На этапе пре-процессинга проводится работа с очисткой данных: обработка пропущенных значений, выбросов, дублей, проверка целостности и логичности данных.

Примеры обработки:

Пропущенные значения (например, 10, 20, NULL, 0) - постановка:

* Для непрерывных – среднее, медиана, значение
* Для категориальных – мода, значение, кодировка

Для выбросов (например, 10, 15, 20, 1000000) – 1% квантиль либо выше допустимого значения:

* Замена на 1% (или прочую) квантиль
* Замена на среднее значение, медиану, кодировка

Важно: пропущенные значения и выбросы могут коррелировать с сегментом, например, с регионом, с временным интервалом

Дубли – действительно ли это дубли и в чем причина:

* Техническая – связка таблиц БД, ключ
* Объективная – одинаковые анкеты, одинаковый исход

Проверяем наличие дублей. В нашем случае количество дублей равно 0.

Исследование и анализ качества данных и распределений.

Пример распределений по числовым характеристикам

# > summary(df$NetIncome)

# Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

# 0 1950 2787 5173 4000 90044313 36

# > summary(df$Age)

# Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

# -1.00 29.00 37.00 38.83 48.00 94.00 1

Пример распределений по категориальным характеристикам

# > summary(df$Marital.status)

# Вдова Женат Замужем Не замужем Разведен разведена Сожитель Холост NA's

# 8112 99528 79208 14012 3571 4504 226 26262 275

**Таблица 1. Анализ распределений и качества данных для предикторов выборки**



**Выводы по выборке данных для анализа.**

Была использовано выборка данных объемом 235,698 записей.

Количество потенциальных предикторов — 38.

Количество переменных с большим количеством и процентом пропущенных значений (более 50%) небольшое — 9. Для части предикторов пропущенные значения являются информацией и пропущенное значение будет использоваться как значение переменной, атрибут отдельного класса «Пропущенные» при классификации.

**Качество данных.**

**Пропущенные данные.** В вошедших переменных количество пропущенных значений является минимальным, допустимым для того, чтобы объединять их в категории с другими бинами. Пропущенные значения переменной, в случае их небольшого количества, например, не более 10% пропущенных в переменной, добавляются либо к доминирующей категории, либо к наиболее близкой по бизнес-логике.

Поле NetMainIncome. Кол-во пропущенных значений 36. Заменены на минимальную сумму дохода 600. Эта замена в дальнейшем не имеет значения, поскольку при бининге нижний интервал значительно выше (3200) и переменная вошла только в одну модель.

Поле SourceMainIncome. Пропущенные значения всего составляют 1064 значения или 0.45%, и вошли в категорию Прочие.

Поле Education. Missing Rate = 5%. Вошла в одну из скоркарт как отдельная категория. Рекомендации по обработке указаны - либо по наименьшему баллу, либо по среднему, поскольку причина такого заполнения неизвестна.

Поле Marital Status. Missing Rate = 0.12%. Вошла в скоркарты как отдельная категория. Рекомендации по обработке указаны - либо по наименьшему баллу, либо по среднему, поскольку причина такого заполнения неизвестна.

**Выбросы**.

Поле NetMainIncome. Менее 1% значений >10000. Заменены на 10000. Эта замена не имеет значения, поскольку все они объединены в бин для высоких значений дохода.

Age - один случай некорректной даты рождения.

Более выбросов в значимых переменных не наблюдается.

Перечень значимых переменных представлен в разделе 6.

**3. Очистка излишних переменных**

**Убирается ненужные переменные.**

Создается список переменных, которые в дальнейшем используются для анализа

Переменные, которые были убраны сразу, согласно логике.

Например, Branch.ID, Product.ID - нам не нужны ID, Date.of.birth - мы уже использовали дату рождения, City.of.Living - используем регион,

Переменные Collateral.type, Value.of.collateral, Property.type/Collateral.type заменены на переменную-индикатор IsCollateral.

Переменные, которые были убраны после анализа по технически причинам.

Например, Source.of.additional.income - очень много значений, необходимо использовать справочник классификации, но нет фактов, подтверждающих, ценность переменной. Employment.sector - используем сегмент.

Все поведенческие переменные, кроме целевых бинарных переменных "GB\_90Ever", "GB\_CumDlq90", "GB\_60Ever", также исключаются из выборки — для задачи аппликационного скоринга мы будем использовать метрики глубины просрочки.

Переменные, включаемые в выборку для дальнейшего анализа:

"Customer.ID.x",

"Account.ID",

"Loan.Amount",

"IsCollateral",

"Salary.payment.in.bank.account",

"Age",

"Gender",

"Region.of.living",

"Region.of.registration",

"CityOfLivingEqRegistration",

"Education",

"Marital.status",

"Dependants",

"Months.at.current.address",

"Employment.type",

"Employment.segment",

"Months.at.job",

"Net.main.income",

"Source.of.main.income",

"Additional.income",

"Reported.expenses",

"Months.with.bank",

"Client.type",

"Property.object",

"Eskhata.Online",

"Plastic.Cards",

"Deposit",

"GB\_90Ever",

"GB\_CumDlq90",

"GB\_60Ever",

"GB",

"BKI.Rating",

"BKI.Number.of.Loans",

"LoanMonth",

"PreviousLoans\_Count",

"PreviousLoans\_MaxDPD"

Обработка общего набора данных закончена.

**4. Сегментация по видам деятельности**

Разделяется выборка данных на три под выборки согласно сегментам по Типу занятости:

* Работает в организации,
* Собственный бизнес,
* Имеет другой источник дохода.

В данном случае применяется бизнес логика. Данные три сегмента представляют различные схемы получения дохода, следовательно, ожидаются различные модели поведения клиентов и влияние различных наборов риск-драйверов на целевую переменную.

Для каждого из трех сегментов обучается и внедряется своя скоринговая модель. Код разделения на сегменты представлен ниже:

sample\_Empl <- df[df$Employment.type == "Работает в организации"

&

df$Employment.segment %in%

c("Мед. работник",

"Работник в сфере образования",

"Работник госструктур",

"Работник НПО (Ташкилоти Чамъияти)",

"Работник производства",

"Работник сельского хозяйство",

"Работник частной организации",

"Строитель",

"Экономист")

,]

sample\_Bus <- df[df$Employment.type == "Собственный бизнес"

&

df$Employment.segment %in%

c("Агро",

"Производство",

"Торговля",

"Услуги",

"Услуги Мастера",

"Услуги транспорта")

,]

sample\_Other <- df[df$Employment.type == "Имеет другой источник дохода"

&

is.na(df$Employment.segment),]

Для сегмента «Работник организации» используется sample\_Empl.

**5. Выбор целевой переменной и определение подвыборок**

**Определение горизонта прогнозирования для целевой переменной.**

Обычно для определения горизонта прогнозирования целевой переменной может применяться винтажный анализ.

**Винтажный анализ** - кривая зависимости уровня определенной корзины просрочки от месяца на балансе по поколениям выдачи. Выбирается такой период месяцев на балансе (после выдачи кредита), после которого увеличение (прирост) уровня дефолта не происходит значимо, т. е. кривая роста просрочки сходит к какому-либо значению. Такой подход применяется при фиксированном периоде вызревания кредита.

Результатом винтажного анализа может быть решение сократить количество месяцев для критерия дефолта (например, с общепринятых для Базель — 12 месяцев до 9 месяцев), что позволит добавлять в выборку более новые поколения выдачи и строить модель на более актуальных данных.

Результатом винтажного анализа может быть увеличение количество месяцев для периода вызревания кредита (например, с общепринятых для Базель — 12 месяцев до 18 месяцев) с целью более полного включения дефолтов и отражения клиентов, у которых появились проблемы с кредитоспособностью в более поздние периоды, что является характерным для долгосрочных кредитов и периодов финансовых кризисов.

Такой подход является оправданным для портфелей с длительной историей (3 и более лет истории выдач), долгим сроком погашения (2 и более лет), и достаточным количеством наблюдений и дефолтов.

В используемой нами методологии предлагается не ограничивать окно наблюдений для целевой переменной — период вызревания предполагается максимально допустимый.

С одной стороны, это создает неравные условия для поколений выдачи: более ранние кредиты имеют больше шансов выпадения в просрочку и дефолт, чем поздние выдачи.

С другой стороны такой подход позволяет:

1) максимально использовать критерий попадания в корзину просрочки для определения дефолта — пересечение порога уровня дней просрочки в течение всей жизни кредита когда-либо — MAX DPD N+ Ever, и увеличить количество дефолтов;

2) не полагаться на результаты винтажного анализа, критерий сходимости для которого является субъективным и часто варьируется между поколениями выдачи.

Это позволяет повысить дискриминационную силу модели, что является приоритетным критерием для операционной деятельности.

В случае использования модели кредитного скоринга для прогнозирования вероятности дефолта по определенным критериям, например, для задач Basel, IFRS — оценка PD для периода вызревания 12 месяцев, DPD90+@12Month, необходимо произвести калибрацию модели. Калибрация представляет собой линейное преобразование, где каждой оценке вероятности события, используемого при обучении модели, ставится в соответствие наблюдаемая вероятность события по заданному критерию (см. раздел 7 документа).

**Выбранный подход к определению целевой характеристики.**

По причине:

а) низкого уровня дефолта и относительно малого количества наблюдений «плохих» случаев,

б) кратко- и среднесрочного периода выдачи кредитов — преимущественно 12–24 месяца,

в) относительно короткого периода выдач с доступной историей погашений для определения дефолта — максимум 1,5 года

предложено использовать максимально доступный период для определения состояние дефолта кредита — Lifetime («за весь доступный жизненный цикл») или Ever («когда-либо») подход,

В связи с этим винтажный анализ не используется и ограничения по окну наблюдений не применяются. Для прогноза вероятности дефолта согласно заданных критериев рекомендуется использовать процедуру калибрации модели.

**Определение глубины просрочки для целевой переменной**

Проводится анализ распределения целевой переменной по месяцам выдачи кредита.

На основе анализа определим,

1) какую из целевых характеристик: Max DPD 90+, Max DPD 60+, или Cumulative Delinquency > 90 будут использованы как определение дефолта для модели

2) какие периоды будут выбраны для обучающей и тестовых выборок.

Критерий для выбора целевой характеристики: количество событий (плохих договоров) должно быть достаточным для построения модели. Под достаточностью понимаются количество и процент от общего количества.

Например, 20-30 плохих договоров - явно нерепрезентативно для любого объема выборки. 100 - возможно, но для большого количества наблюдений (например, 10 тыс. договоров) дает низкий уровень дефолта и становится сложным либо нерепрезентативным расчет соотношения хороших и плохих наблюдений для отдельных бинов (сегментов значений предикторов). Например, 50 к 500 (~10%) является статистически значимым соотношением, но 5 к 50 не является статистически значимым и дает высокое отклонение (чувствительность) в оценке отношения при изменении количества наблюдений: +1 = 6/50=0.12, -1 = 4/50=0.08.

Чем большее количество "плохих" кейсов, тем большее количество сегментов мы можем использовать. Например, для 50 плохих кейсов разбиение на 3 сегмента может дать 30,15,5, где 5 уже малорепрезентативно, и рекомендуется сократить кол-во сегментов до 2-х.

Критерий для выбора месяца разбиения на обучающую и тестовую (Out-of-time) выборки:

а) в тестовой выборке должно быть достаточное количество наблюдений с событием

б) количество наблюдений должно быть репрезентативным

в) период для тестовой выборки должен включать самые актуальные, последние договора.

Были выбраны критерий «Плохого» клиента, исходя из 1) получения максимально возможного количества наблюдений "плохих" клиентов И 2) соблюдения бизнес-логики определения плохого клиента у Заказчика. Окно вызревания НЕ задается. Критерием хорощий-плохой является пересечения установленного порога DPD N дней метрикой MAX DPD N+ Ever — максимальное количество дней просрочки когда-либо (lifetime).

**Подход к определению тренинговой и тестовой выборок**

Для обеспечения временной стабильности модель разрабатывается и тестируется на различных периодах. Обозначим обучающую выборку Dev, тестовую выборку из другого временного интервала — Out-of-time (OOT).

Период для обучающей выборки (dev) помечен зеленым цветом — 202210 — 202209 — и равен 12 месяцам, период для тестовой временной выборки (oot) - желтым.

Договора после 202203 не входят в анализ, поскольку для них нет достаточного времени для реализации события дефолта.

Период для out-of-time выбран таким образом, чтобы захватить период от «+12 месяцев от начала обучающей выборки» до максимально возможно близкого к дате формирования выборки.

**Таблица 2. Распределение целевых характеристик по месяцам выдачи для сегмента «Работает в организации»**



Выбираем вариант GB\_60Ever\_Month - паритет достаточного кол-ва событий и привязанности критерия к индикатору плохого кредита для отчетности и резервирования.

Создаем переменную GB и присваиваем ей значения из GB\_60Ever.

**6. Обучающие и тестовые выборки (семплинг)**

Семплинг – это создание выборок для обучения и тестирования.

Development Sample - выборка, на которой будем обучать модель

dev <- sample %>% filter(LoanMonth >= 202110 & LoanMonth < 202210) %>% mutate(LoanMonth = NULL) %>% as.data.frame()

Out-of-time - выборка, на которой будем тестировать модель на временную устойчивость и способность прогнозировать на последних доступных периодах

oot <- sample %>% filter(LoanMonth >= 202210 & LoanMonth < 202303) %>% as.data.frame()

Out-of-time 2 - выборка, на которой будем тестировать модель на временную устойчивость и способность прогнозировать на периоде внутри обучающей выборки. Фактически, реализация кросс-валидации в ручном режиме. Можем менять диапазоны для этого периода множество раз

oot2 <- sample %>% filter(LoanMonth >= 202207 & LoanMonth < 202210) %>% as.data.frame()

Development Sample - выборка, на которой будем обучать финальную модель для внедрения на продакшн ПОСЛЕ тестирования на временную устойчивость. Делается для того, чтобы не терять информацию из out-of-time выборки, и добавить ее в модель

dev\_final <- sample %>% filter(LoanMonth >= 202110 & LoanMonth < 202302) %>% mutate(LoanMonth = NULL) %>% as.data.frame()

Количество наблюдений в выборках:

Development = 27834

OOT = 12429

Development Final = 37636.

Development Final используется для обучения финальной модели после проверки на стабильность.

Распределение на Train и Test для Модели 1

|  |  |  |
| --- | --- | --- |
| Development | 27834 | DR=2.15% |
| Train \* 0,67 | 18588 |  |
| Test \* 0.33 | 9246 |  |

Всего получаем 5 дата семплов,

dev = train + test (out-of-sample) — обучающая выборка

oot

dev\_final = dev + oot (out-of-time) — полная выборка, становится обучающей на финальном этапе

# ПРИЛОЖЕНИЕ №5. МОДЕЛИРОВАНИЕ. ДЛЯ «САМОЗАНЯТЫЕ»

Процесс моделирования представляет собой 4 этапа:

Предварительная группировка интервальных параметров в корзины по 5% (может быть другое значение)

Биннинг с оптимальным разбиением на категории с соблюдением баланса между количеством групп, значимостью групп (Information Value) с ограничением на бизнес-логику. Расчет WoE для каждого бина.

Выбор переменных в модель согласно IV и бизнес логике.

Регрессионный анализ — логистическая регрессия, где целевая характеристика 0/1 и предикторы — WoE, соответствующие значениям переменных в соответствующих диапазонах бинов.

Возможно также применение Dummy (Binary) Variables (булевых или фиктивных переменных), используя Encoding, когда для каждой переменно создается с N бинов создается N-1 переменных, принимающие значения 0 или 1. Я рекомендую использовать метод с заменой на WoE, т. к. он позволяет сохранить в логистической регрессии и скоринговой карте тренды, ожидаемые на этапе биннинга по бизнес-логике предикторов.



**1. Выбор значимых переменных**

Начинается построение скоринговой модели.

Рассчитывается метрика Information Value на первоначальных значениях переменных. Цель - отсеять то, что априори не является значимым.

Мера вероятности «хорошего» исхода в данной характеристике – соотношение доли «хороших» в атрибуте к доле «плохих» исходов



pi = доля «хороших», имеющих атрибут i, во всех хороших

qi = доля «плохих», имеющих атрибут i, во всех плохих

и



Weight of Evidence атрибута характеристики:



Information Value переменной:



Функция iv возвращает таблицу со списком предикторов и значениями Information Value для целевой GB.

iv\_values\_final <- iv(dev\_final, y = "GB")

Отбрасывается переменные со значением информационной значимости менее 0.02 и количеством пропущенных значений более 95%

Удаляем переменные — ID клиента, договора, индикаторы хороших/плохих договоров, они нам больше не нужны.

Выборку формируем на основе семпла dev.

В результате остается 21 переменных.

**Таблица 3. Information Value предикторов**



**2. Биннинг переменных**

Проводится трансформация переменных в категориальный формат - биннинг (coarse classing)

Применяется алгоритм оптимального разбиения на классы с ограничением по указанным критериям. Критерий оптимизации хорошо работает для интервальных переменных, но не является оптимальным для категориальных переменных.

Под оптимальным разбиением подразумевается найти такие диапазоны и количество классов, которые давали бы максимальное IV переменной с учетом ограничений на кол-во бинов и значимость.

breaks = list(

Region.of.living = c(

"Вилояти Сугд",

"Вилояти Хатлон%,%ВМКБ",

"Душанбе",

"Нохияхои тобеи Чумхури"),

Marital.status = c(

"Женат%,%Сожитель",

"Замужем%,%Вдова%,%разведена",

"Холост",

"Не замужем%,%Разведен"

),

Source.of.main.income = c(

"Доход семьи",

"Зарплата по основному месту работы",

"Предпринимательство",

"Пенсия%,%Прочее%,%missing"

),

Months.at.job = c(61,109,181),

Months.with.bank = c(10,25,61)

)

Запускается алгоритм биннинга и расчета Woe (Weight of Evidence) для каждой переменной в выборке по целевой GB.

Используется следующие настройки:

# method = "tree", (Метод оптимизации биннинга, возможно использовать Chi-squared)

# breaks\_list = breaks, (подключаем списко отработки вручную)

# # save\_breaks\_list = "break\_list\_woe5", (можем сохранить диапазоны разбиения в отдельном файле)

# check\_cate\_num = FALSE, (проверять кол-во разных значений в категориальных переменных. Например, если более 50 разных значений, как-то города, можно пропустить для экономии времени)

# count\_distr\_limit = 0.05, (Доля минимальное количество наблюдений в бине от общего количества наблюдений. В данном случае, размер бина не может быть менее 5% от популяции)

# bin\_num\_limit = 5 (Максимальное количество бинов. В данном случае, мы используем не более 5. Для данной выборки, дефолт рейта и кол-ва плохих кейсов может использоваться и меньшее число. Это делат модель менее точной, НО более стабильной)

Цель биннинга WoE - заменить оригинальные переменные на значения WoE. Так мы получаем:

1) возможность работать только с категориальным переменными, но подавать на вход модели числовое значение, являющееся мерой отклонения соотношения "Хороших" и "Плохих" наблюдений в бине от соотношения "Хороших" и "Плохих" наблюдений в выборке в целом.

2) избавляет от необходимости объяснения возможных нелинейных трендов. Линейная модель логистическая получает возможность обрабатывать нелинейные зависимости.

3) избавляет от необходимости обработки пропущенных значений и выбросов (если это не было сделано отдельно специальным образом согласно бизнес-логике). Возможно работать с категорией пропущенных значений ка с отдельной категорией, либо включить ее наиболее подходящий бин.

Пример вывода результатов функции woebin по некоторым переменным (которые вошли в дальнейшем в модель).

Variable — имя переменной

count — количество наблюдений в БИНЕ

count\_dist — часть от популяции в БИНЕ

neg — количество «негативных», т. е. «Хороших» - негативный тест на дефолт

pos - количество «позитивных», т. е. «Плохих» - позитивный тест на дефолт

posprob — Default Rate — доля плохих в БИНЕ

woe — значение Weight of Evidence

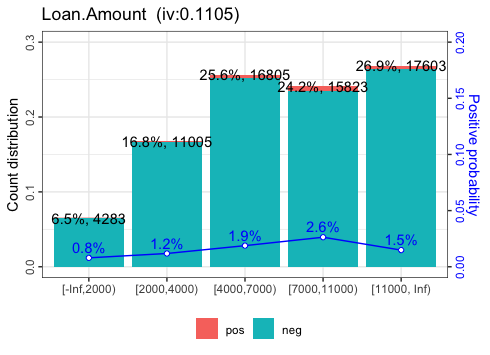
bin\_iv — Information Value для бина

total\_iv — Information Value для переменной

**Таблица 4. Результаты биннинга переменных (представлены, вошедшие в последствие в модель)**



Примеры вывода графиков WoE для визуального анализа

Изображение выглядит как текст, снимок экрана, диаграмма, График

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, График, диаграмма

Автоматически созданное описание

**3. Финальная подготовка переменных и выборок для анализа**

Разбиение тренинговой выборки на 2 семпла: Train и Test (Out-of-sample) для анализа структурной устойчивость модели.

Здесь применяются пропорции 1:2, 0.67 к 0.33.

Применяем полученные бины и значения WoE для всех выборок.

Будем использовать значения WoE каждого диапазона категориальных переменных как предикторы модели.

Фактически мы заменяем интервальные (числовые) переменные как то возраст, доход и прочее на категории "от - до" для ухода от нелинейных и неустойчивых из-за разного кол-ва наблюдений зависимостей между предиктором и ln(p/1-p) , а категориальные группируем в более крупные группы для устойчивости и статистической значимости.

Рассчитываем Information Value для преобразованных переменных. Реальный IV для предикторов в модели. Можем отбросить переменные с низким и очень высоким IV в автоматическом режиме.

**Таблица 5. Матрица корреляции переменных**



Сильно коррелирующими переменными (коэффициент корреляции более 0.67, на практике для скоринга — более 0.4) здесь являются:

Регион проживания и Регион регистрации;

Предыдущее количество кредитов и Предыдущая максимальная просрочка.

Мы их НЕ исключаем из выборки на данном этапе, поскольку нет понимания, какая из переменных даст большую ценность для финальной модели. Исключение производится при анализе значимости оценок логистической регрессии.

**4. Обучение и выбор модели**

Для тестирования стабильности модели предлагается следующий подход. Фактически он реализует идею кросс-валидации, только делает акцент на том, как можно протестировать и скорректировать модель для достижения баланса между точностью дискриминации и стабильностью модели.

Обращаете внимание, что для скоринга, как для задачи бинарной классификации, важно в первую очередь качество разделения наблюдений на два класса. Точность прогноза вероятности события на уровне договора или клиента как правило вторична и достигается калибрации модели и обеспечением монотонности зависимости между оценкой риска и реальным наблюдаемым уровнем дефолта.

Общий набор данных dev\_final разделен на следующие выборки:

Обучающая выборка dev, состоит из тренинговой (train) и тестовой структурной (test out-of sample).

Тестовая временная выборка (test out-of-time).

dev = train + test (out\_of\_sample)

dev\_final = dev + oot

Моделирование и выбор модели проводим по следующему алгоритму:

1. Запуск логистической регрессии со всеми предикторами на выборке train

# ln(p/1-p) = b0 + b1\*x1 + b2\*x2 + ... + bn\*xn

# p = 1/(1+exp(-1\*(b0 + b1\*x1 + b2\*x2 + ... + bn\*xn)))

2. Запуск метода stepwise для отсеивания незначимых переменных.

Существует три метода автоматического отбора параметров: forward, backward, stepwise.

Forward - в модель добавляются по одному предикторы до тех пор, пока критерий, используемый при решении оптимизационной задачи (AIC), не перестает увеличиваться.

Backward использует обратный подход - в модель подаются все переменные, далее убираются по одной до тех пор, пока критерий оптимизации не начинает уменьшаться.

Метод Stepwise использует смешанный подход - добавление сильных и удаление слабых переменных, и дает оптимальный набор предикторов.

Может быть ресурсоемким. Для выборки на 100 тыс. наблюдений и 100 предикторов может занять до нескольких часов.

3. Отбор оставшихся параметров логистической регрессии в ручном режиме

4. Пересчет параметров валидации для каждой итерации (каждого нового набора переменных) на выборках train, test, oot.

Этапы 3–4 повторяются вместе итерационно.

AUC не должен отличаться существенно для train, test, oot. Убираем незначимые, слабо значимые, переменные с Pr(>|z|) > 0.01 до тех пор, пока:

1) AUC train не сойдется с AUC oot,

НО

2) AUC train и AUC oot не начнут значимо падать.

Такой баланс удастся соблюсти итеративным опытным путем. Процесс относительно творческий, как и биннинг переменных, поэтому единого детерминированного подхода "как получить самую лучшую модель" нет. Есть баланс между точностью и стабильностью.

Получена стабильная наилучшая модель на тренинговой выборке и оттестирована на под выборке из другого временного промежутка. Теперь нужно использовать максимально данные выборки для включения самых последних данных. Добавляем OOT в обучающую выборку.

5. Запуск логистической регрессии со всеми предикторами на выборке dev\_final.

6. Запуск метода stepwise для отсеивания незначимых переменных.

7. Отбор оставшихся параметров в ручном режиме

8. Пересчет параметров валидации для каждой итерации (каждого нового набора переменных) на выборках dev\_final, oot, oot2 (выборка из начального временного интервала).

Этапы 7–8 повторяются вместе итерационно.

Выборку oot2 можно переопределять по различным периодам, как это происходит во время процесса автоматизированной кросс-валидации.

Таким образом мы смотрим, как модель, построенная на общем, предсказывает на частных под выборках.

9. Сравниваем набор параметров из модели TRAIN с набором параметров из модели DEV\_FINAL. Наборы не должны отличаться, либо отличаться на 1-2 переменные, включение или исключение которых мы должны проанализировать и объяснить. Такая разница может свидетельствовать о временной неустойчивости прогнозной силы и/или значимости отдельных переменных.

10. Финализируем модель на выборке DEV\_FINAL.

***Проводится обучение модели логистической регрессии на данных выборки train***

Вывод результатов оценки коэффициентов логистической регрессии после применения метода Stepwise.

Убирается те переменные, для которых в выводе регрессии высокие значения chi-squared (p-level, t-value), более 0.1 - однозначно выбрасываем, больше 0.05 - смотрим, нужны ли они нам, более 0.01 также можем оставлять, если переменных большое кол-во, более 10 штук в модели.

Данный процесс проводим последовательно с расчетом метрик прогнозной силы модели на train, test, out-of-time выборках. Значения не должны отличаться существенно.

Если значения отличаются существенно, например, AUC Train = 0.78, AUC OOT = 0.68 - в модели есть незначимые оценки, т. е. оценки с большой дисперсией (гетероскедастичность). Мы получаем хороши результаты на тренинговой выборке, но плохие на тестовой из другого временного интервала.

Сокращаем кол-во предикторов до тех пор, пока AUC Train и AUC OOT не сойдутся до разницы в 0.01–0.03.

# Output

# Call:

# glm(formula = final\_formula\_train, family = binomial, data = train\_woe)

#

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -3.78879 0.05120 -73.993 < 2e-16 \*\*\*

Loan.Amount\_woe 0.99107 0.12515 7.919 2.39e-15 \*\*\*

Region.of.living\_woe 0.30822 0.08943 3.447 0.000568 \*\*\*

Marital.status\_woe 0.16166 0.08999 1.796 0.072425 .

Employment.segment\_woe 0.61566 0.08608 7.152 8.53e-13 \*\*\*

Net.main.income\_woe 0.52886 0.17695 2.989 0.002802 \*\*

Source.of.main.income\_woe 0.51936 0.19964 2.601 0.009284 \*\*

Months.with.bank\_woe 0.44507 0.07626 5.836 5.33e-09 \*\*\*

BKI.Rating\_woe 1.02096 0.12879 7.928 2.23e-15 \*\*\*

Dependants\_woe 0.48344 0.16690 2.897 0.003771 \*\*

Months.at.job\_woe 0.27811 0.10792 2.577 0.009969 \*\*

Age\_woe 0.42751 0.08980 4.761 1.93e-06 \*\*\*

PreviousLoans\_MaxDPD\_woe 0.77477 0.14673 5.280 1.29e-07 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Все коэффициенты модели значимы. Сокращать количество предикторов далее смысла не имеет. Значимость предиктора свидетельствует о допустимой дисперсии оценки и отсутствии гетероскедастичности.

***Валидация модели.***

Запускается расчет прогнозных вероятностей события для подвыборок - обучающая, тестовая out-of-sample, тестовая out-of-time.

oot\_perf\_train — валидационные метрики при применении модели, построенной на выборке train к дата семплу oot.

В выводе мы видим, что AUC для train, test, oot отличаются незначительно, на ~1%.

Переобучения модели (overfitting), т.е. подгонки параметров под обучающий набор данных, не наблюдается.

Небольшая разница в AUC для train, test, oot свидетельствует о стабильности модели и параметров, используемых в модели.

> # Выводим набор метрик для указанных подвыборок.

> train\_perf$binomial\_metric

$`M2 - Business - Train`

MSE RMSE LogLoss R2 KS AUC Gini

1: 0.02461712 0.1568984 0.1079789 0.03484008 0.4083586 0.7731005 0.546201

> test\_perf$binomial\_metric

$`M2 - Business - Test`

MSE RMSE LogLoss R2 KS AUC Gini

1: 0.02427646 0.1558091 0.1079093 0.02901385 0.4022266 0.7581434 0.5162868

> oot\_perf\_train$binomial\_metric

$`M2 - Business - OOT with Train`

MSE RMSE LogLoss R2 KS AUC Gini

1: 0.007354961 0.08576107 0.04321983 -0.02378755 0.3760453 0.7459799 0.4919597

***Проводится обучение модели логистической регрессии на полных данных: dev sample + oot sample = dev\_final.***

После метода Stepwise и ручного отбора незначимых параметров И параметров, которые не согласуются с моделью на семпле train получаем следующую модель:

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -4.02349 0.03468 -116.027 < 2e-16 \*\*\*

Loan.Amount\_woe 0.82952 0.09153 9.063 < 2e-16 \*\*\*

Region.of.living\_woe 0.34516 0.06785 5.087 3.64e-07 \*\*\*

Marital.status\_woe 0.34671 0.06571 5.277 1.32e-07 \*\*\*

Employment.segment\_woe 0.61584 0.05642 10.916 < 2e-16 \*\*\*

Net.main.income\_woe 0.53208 0.12978 4.100 4.14e-05 \*\*\*

Source.of.main.income\_woe 0.46240 0.14916 3.100 0.00194 \*\*

Months.with.bank\_woe 0.65248 0.05611 11.629 < 2e-16 \*\*\*

BKI.Rating\_woe 0.80484 0.09023 8.920 < 2e-16 \*\*\*

Dependants\_woe 0.38250 0.12127 3.154 0.00161 \*\*

Months.at.job\_woe 0.23858 0.07939 3.005 0.00265 \*\*

Age\_woe 0.39656 0.06622 5.989 2.12e-09 \*\*\*

PreviousLoans\_MaxDPD\_woe 0.86651 0.08665 10.000 < 2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

***Валидация модели.***

dev\_perf — валидационные параметры на всех периодах с моделью построенной по всем периодам

oot\_perf — валидационные параметры на периодах oot с моделью построенной по всем периодам

oot2\_perf — валидационные параметры на периодах oot2 (подвыборка из выборки dev\_final) с моделью построенной по всем периодам

***Сравниваем выводы и метрики валидации моделей 1–4 и 5-8.***

Финальная модель. Полученные коэффициенты логистической регрессии и их значимость немного отличаются от полученных на Этапе 3–4 на выборке Train.

> dev\_perf$binomial\_metric

$`M2 - Business - Dev`

MSE RMSE LogLoss R2 KS AUC Gini

1: 0.01683358 0.1297443 0.07844789 0.02960609 0.4336531 0.7881279 0.5762558

> oot\_perf$binomial\_metric

$`M2 - Business - OOT`

MSE RMSE LogLoss R2 KS AUC Gini

1: 0.007220363 0.08497272 0.04106552 -0.005051812 0.3851178 0.7532391 0.5064782

> oot2\_perf$binomial\_metric

$`M2 - Business - OOT2`

MSE RMSE LogLoss R2 KS AUC Gini

1: 0.02258862 0.1502951 0.1013732 0.02587317 0.4074233 0.7649714 0.5299428

> train\_perf$binomial\_metric

$`M2 - Business - Train`

MSE RMSE LogLoss R2 KS AUC Gini

1: 0.02461712 0.1568984 0.1079789 0.03484008 0.4083586 0.7731005 0.546201

> test\_perf$binomial\_metric

$`M2 - Business - Test`

MSE RMSE LogLoss R2 KS AUC Gini

1: 0.02427646 0.1558091 0.1079093 0.02901385 0.4022266 0.7581434 0.5162868

> oot\_perf\_train$binomial\_metric

$`M2 - Business - OOT with Train`

MSE RMSE LogLoss R2 KS AUC Gini

1: 0.007354961 0.08576107 0.04321983 -0.02378755 0.3760453 0.7459799 0.4919597

**Таблица 6. Сравнительный анализ метрика прогнозной силы модели на разных выборках**



Возьмем за основу метрику AUC. OOT семпл на полной модели показывает результаты не ниже Train и DevFull — 75.32%. В реальности мы должны ожидать, что модель покажет результат AUC OOT Train Model = 74.60. Это не отличается существенно от обучения и теста модели, построенной на полных данных (включая oot период) — 77.3% и 75.8%. Модель стабильная и показывает хорошую прогнозную силу.

***Финальная модель. Сохраняется коэффициенты регрессии.***

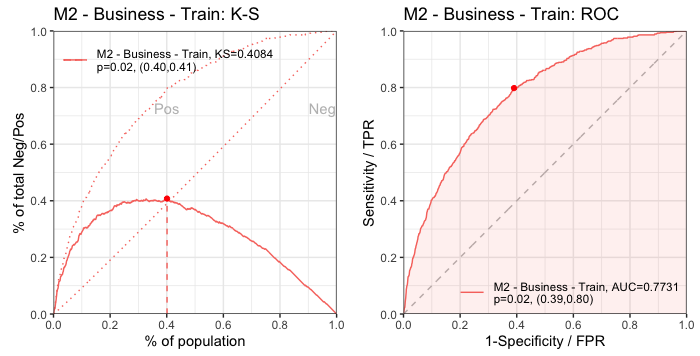
Дальнейшее использование при внедрении — коэффициенты регрессии умножаются на значения WoE соответствующего бина переменной.

**Таблица 7. Финальные оценки параметров логистической регрессии**



**5. Финальные отчеты по валидации модели**

Результаты этапа 4. Валидация модели, построенной на выборке Train, представленные в виде графиков



Изображение выглядит как текст, диаграмма, График, линия

Автоматически созданное описание

Изображение выглядит как текст, диаграмма, линия, График

Автоматически созданное описание

Результаты этапа 8. Валидация модели на ВСЕХ периодах выборки dev\_final представленные в виде графиков

Изображение выглядит как текст, диаграмма, линия, График

Автоматически созданное описание

Изображение выглядит как текст, диаграмма, линия, График

Автоматически созданное описание

Изображение выглядит как текст, диаграмма, График, линия

Автоматически созданное описание

Матрица неточностей (матрица неточностей) — это таблица, которая демонстрирует эффективность алгоритма классификации путем сравнения прогнозируемого значения целевой переменной с ее фактическим значением.

Матрица Confusion Matrix представлена информативно для threshold, обеспечивающего максимальное значение F1-score, т. е. дающее минимальное гармоническое произведение ошибок 1-го и 2-го рода. Для принятия решений балл отсечения смещается согласно стратегии для обеспечения желаемых показателей бизнес стратегии.



Показатель F1-score является гармонически средним показателей Precision — доля правильно угаданных из спрогнозированных плохих и Recall — доля правильно угаданных из реальных плохих. Его значение зависит от выбранного балла отсечения. В данном случае подбирается такой балл, который дает максимальное значение F1-score для данной выборки за счет баланса между Precision и Recall показателями, т. е. ошибками 1-го и 2-го рода. Однако, невысокие показатели обусловлены высокой несбалансированностью выборки (малый вес событий в общем количестве наблюдений). Увеличивая отсечение по плохим договорам (снижая ошибку 2-го рода — уменьшая уровень просрочки, мы повышаем ошибку первого рода — снижаем уровень выдач. Общая ошибка модели составляет всего 3.65%, т. е. мы точно предсказали 96% всех исходов.

Гетероскедачность. Не применяется МНК (линейную регрессию), поэтому даже наличие гетероскедастичности - зависимости дисперсии ошибок от переменной, не является критичным. Более того, в задачах бинарной классификации (биномиальное распределение) нарушается предположение о гомоскедастичности и ошибки не распределены нормально. Теоретически, мы можем проверять, например, "увеличивается ли соотношение 1/0 при увеличении суммы дохода", но это не повлияет на наш выбор: если переменная имеет высокое или низкое Information Value, если оценки коэффициентов регрессии значимая или незначимая - мы переменные дающие высокую гетероскедастичность отбрасываем на этом этапе. Более того, применяется метод биннинга - трансформации в категориальные переменные, а значит зависимость между переменной и дисперсией отклонений фактически отсутствует.

Расчет устойчивости переменных и популяции. Коэффициент PSI (Population Stability Index) представлен в каждом отчете - сравнение под выборки Train и Out-of-Time семплов. Значения от 0.02 до 0.05, что говорит о довольно высокой стабильности популяции. Однако, не следует путать стабильность популяции со стабильностью прогнозной силы модели. Нестабильная популяция не обязательно свидетельствует о низкой прогнозной силе модели, и наоборот, стабильная популяция может давать дискриминацию близкую к нулю. Мы проверяем устойчивость переменных двухэтапным процессом отбора переменных: а) исключаются переменные с незначимыми коэффициентами регрессии на train sample - устранение мультиколлинеарности, б) исключаются дополнительно переменные с незначимыми коэффициентами регрессии на общем development sample

# ПРИЛОЖЕНИЕ №6. СКОРИНГОВАЯ КАРТА И КАЛИБРАЦИЯ. «САМОЗАНЯТЫЕ»

Полученные коэффициенты логистической регрессии и значения WoE могут применяться для расчета оценки вероятности дефолта. Для параметров регрессии b0, b1,…, bn и характеристик (переменных) x1,…,xn вероятность дефолта рассчитывается по формуле:

Значения скоринга (балла, функции регрессии) в этом случае обычно приобретают значения порядка от -5 до 5 с большим количеством знаков после запятой, что не всегда удобно для презентации и принятия решений. Для удобства принято использовать баллы, приведенные к определенной шкале, округленные и укрупненные.

Баллы скоринговой карты получаем согласно шкалы FICO. Можно использовать любую свою удобную шкалу. Для параметра i некалиброванный балл получается, как произведение коэффициента логистической регрессии параметра и WoE для соответствующего бина:

RawScore i = Estimate (LogRegr)i \* WoEi.

Подробно процедура калибрации и расчёта вероятности дефолта для скорингового балла изложена в разделе «Калибровка системы показателей и присвоение оценок».

Настраиваемые параметры — значения даны согласно шкалы FICO:

points0 = 660 - Балл, для которого выполняется соотношение odds (условное начало координат)

odds0 = 1/72 - соотношение odds плохих к хорошим. Здесь - на одного плохого приходится 72 хороших случая, в точке 660.

pdo = 40 - шаг, при котором соотношение odds плохих к хорошим увеличивается или уменьшается в 2 раза.

вероятность дефолта для калиброванного балла может быть рассчитана как:

**Таблица 8. Скоринговая карта**



**Калибровка системы показателей и присвоение оценок**

Необработанные оценки из регрессионных моделей обеспечивают ранжирование клиентов с точки зрения дефолта, но оценки нельзя сравнивать с оценками, полученными из другой системы показателей. Калибровка стандартизирует взаимосвязь между баллами и PD, позволяя напрямую сопоставить баллы из разных систем показателей. Банк может иметь много систем показателей с различными шкалами и без последовательной связи PD-score. Затем калибровка необходима для определения оценочных классов, используемых для объединения, с точки зрения единой стандартной шкалы. В более общем плане калибровку можно рассматривать как процесс присвоения вероятности дефолта (PD) рейтинговым оценкам.

Процесс калибровки выполняет следующие 2 этапа:

* Установление взаимосвязи между внутренними баллами и коэффициентами;
* Определение оценок и связанных с ними PD.

**Стандартная шкала**

Стандартная шкала определяется путем указания связи между баллами и Хорошим/Плохим

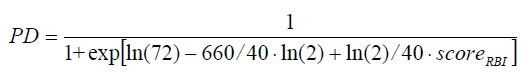
коэффициенты (или PD). Стандартная шкала определяется на основании следующих правил:

* Используя отраслевой стандарт FICO 72:1 коэффициент 660.
* Каждые 40 очков прибавки в счете удваивают шансы.

**Таблица 9. Шкала FICO**

|  |  |  |  |
| --- | --- | --- | --- |
| Score | GB Odds | ln(Odds) | PD |
| 460 | 2,25 | 0,81093 | 0,30769 |
| 500 | 4,5 | 1,50408 | 0,18182 |
| 540 | 9 | 2,19722 | 0,10000 |
| 580 | 18 | 2,89037 | 0,05263 |
| 620 | 36 | 3,58352 | 0,02703 |
| **660** | **72** | **4,27667** | 0,01370 |
| 700 | 144 | 4,96981 | 0,00690 |
| 740 | 288 | 5,66296 | 0,00346 |
| 780 | 576 | 6,35611 | 0,00173 |
| 820 | 1152 | 7,04925 | 0,00087 |
| 860 | 2304 | 7,74240 | 0,00043 |
| 900 | 4608 | 8,43555 | 0,00022 |
| 940 | 9216 | 9,12870 | 0,00011 |

Соотношение между стандартным баллом и PD составляет



Линейный коэффициент для стандартной шкалы можно найти из уравнения:



Таким образом, более простым способом вероятность дефолта для определенного балла может быть рассчитана как:

Форма зависимости вероятности по умолчанию от калиброванной оценки:



Калиброванная оценка представляет линейное преобразование необработанного балла:

Коэффициенты для калиброванного балла можно найти из соответствующего ln(коэффициента) для реальных данных и стандартной шкалы



Калиброванная оценка используется для оценки вероятности события. Коэффициенты 413.203 и 57.7078 являются постоянными для данной шкалы FICO:

Изображение ночного неба

Описание генерируется автоматически

Для других шкал коэффициенты преобразования калиброванного балла в вероятность дефолта и преобразования необработанного балла в калиброванный балл должны быть скорректированы.

Калибровка скоринговой модели проводится периодически на постоянной основе – ежеквартально или раз в полгода, в зависимости от объемов кредитования и изменения вероятности дефолта на уровне пула.

# ПРИЛОЖЕНИЕ №7. БИЗНЕС-СТРАТЕГИИ

**Распределение популяции по скоринговому баллу**

Для использования скоринговой модели, установки баллов отсечения и расчета стратегий нам необходимо проскорить выборку для анализа. Для выборки данных рассчитывается для каждого договора вероятность дефолта и некалиброванный балл - значение коэффициента регрессии переменной, помноженное на значение WoE параметра плюс свободный член.

Применяем полученную скоринговую карту к выборке данных и рассчитываем калиброванный балл, т. е. балл согласно шкале. Добавляем вектор с рассчитанным калиброванным баллом к общей выборке.

Та же самая операция для выборки dev\_final. Это нам нужно для расчета распределений выборки по скоринговому баллу и бэктестинга - логической проверки насколько логично модель определяет уровень дефолта.

Используем функцию gains\_table для получения распределений кол-ва наблюдений и дефолт рейт (вероятности события) по диапазонам скорингового балла.

Важно - Positive — это определение события, для дефолта. Таким образом, Positive = 1 подразумевает позитивный тест на событие (дефолт), но НЕ хороший (позитивный) договор

**Таблица 9. Распределения скорингового балла (для Dev выборки)**



Таблица распределения является инструментом для установления баллов отсечения по стратегии:

1) уровень одобрения (approval rate) - столбцы ApprovalRate, Cum\_Count,

2) уровень дефолта (Cum\_PosProb).

Стратегической кривой называется зависимость (графическая или табличная) между уровнем одобрения заявок (Approval Rate) и прогнозным кумулятивным уровнем дефолта (CumPosProb). Двигаясь по ней, мы можем выбирать варианты балла отсечения.

**Бэк-тестинг модели**

С помощью распределения договоров, хороших и плохих, по скоринговому баллу, можем провести бэктестинг скоринговой модели.

С ростом скорингового балла ожидаемая вероятность дефолта должна уменьшаться. В таблице в столбце PosProb (наблюдаемый уровень дефолта) и Cum\_PosProb (кумулятивный уровень дефолта) значения монотонно убывают с увеличением скорингового балла.

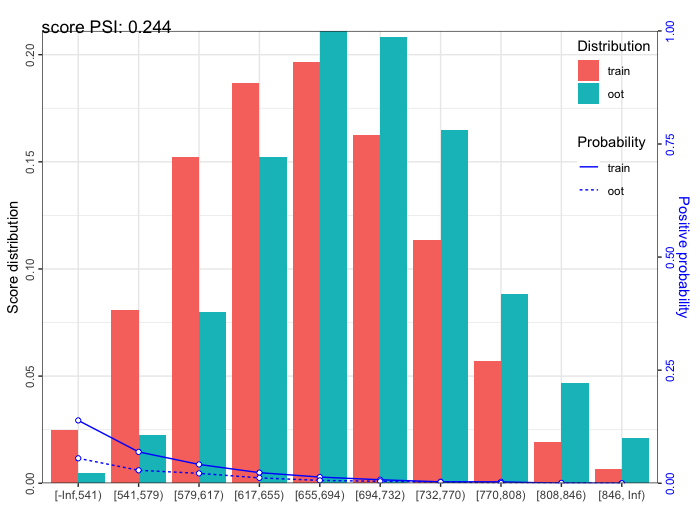
Так, например, в приведенной таблице заявки отсортированы по возрастанию скорингового балла, и ожидается снижение наблюдаемого уровня дефолта с увеличением балла. Наблюдаемый уровень дефолта портфеля (столбец CumPosProb) снижается монотонно относительно роста скорингового балла.

**Стабильность популяции**

Тест стабильности популяции.

Тест стабильности популяции показывает смещение распределения скорингового балла тестируемой популяции относительно базовой.

Для оценки стабильности популяции применяется метрика Population Stability Index (PSI).



Сравнивается распределение популяции обучающей выборки (dev) и временной тестовой выборки (oot) по скоринговому баллу с помощь метрики PSI (Population Stability Index). Значение 0.244 свидетельствует от незначительном смещение профайла клиента весной 2023 года по сравнению с 2022 годом.

# ПРИЛОЖЕНИЕ №8. ВЫВОДЫ

В результате проекта была разработана система кредитного скоринга для розничного портфеля с использование традиционной методологии построения скоринговых карт. Методология заключается в а) трансформации исходных переменных в категории и использование значений Weight of Evidence категорий в качестве входных параметров, б) использовании логистической регрессии в качестве алгоритма решения задачи бинарной классификации, в) представлении модели в качестве калиброванного инструмента балльной оценка — скоринговая карта. Данный метод является робастным, позволяет нивелировать проблемы процессинга данных, связанные с невысоким качеством данных, такие как, например, пропущенные значения и выбросы, и увеличить скорость разработки без потери качества.

Система оценки рисков состоит из трех моделей. Разделение на сегменты произведено согласно критерию источника дохода — “Работник организации”, “Самозанятые” и “Получатели денежных переводов” . Показатели дискриминационной силы скоринговых карт — AUC ~ 0.75–0.80 — имеют значения выше среднего для моделей аппликационного скоринга, что позволяет использовать их для принятия решений в процессе кредитования. Одним из вызовов в процессе построения модели является а) высокая несбалансированность выборки — доля событий (вероятности дефолта) составляет для разных сегментов ~1%-2% от общего количества наблюдений, б) нестабильность (снижение) уровня вероятности дефолта во времени. Это связано с высоким качеством портфеля и снижением уровня просрочки в конце наблюдаемого периода, что также объясняется небольшим периодом для реализации дефолта. При обучении и выборе финальной модели был применен метод ручной кросс-валидации, тестирования на различных временных под выборках, что позволяет достичь баланса между предиктивной силой и стабильностью модели.

Предложенные скоринговые модели построены по технологии, позволяющей учесть и нивелировать возможные негативные факторы данных.

1) Стабильность переменных проверяется с помощью

i) валидации модели на различных временных промежутках,

ii) двухэтапным процессом отбора переменных:

а) исключаются переменные с незначимыми коэффициентами регрессии на train sample - устранение мультиколлинеарности,

б) исключаются дополнительно переменные с незначимыми коэффициентами регрессии на общем development sample.

Это также позволяет нивелировать гетероскедастичность: если есть зависимость между временем и дисперсией, она проявится при оценке параметров регрессии.

2) Подход на основе биннинга как интервальных, так и категориальных переменных позволяет:

а) учесть нелинейные тренды (фактически переходим к категориальной модели),

б) пропущенные значения переменной, в случае их небольшого количества, например, не более 10% пропущенных в переменной, добавляются либо к доминирующей категории, либо к наиболее близкой по бизнес-логике.

3) Основной акцент делается на способность модели прогнозировать на различных временных промежутках, в т. ч. на тех, на которых модель не обучалась (кросс-валидация). Акцент здесь проводится на соблюдении баланса между прогнозной силой модели и стабильностью, избежаний переобучения, когда модель хорошо предсказывает на тренинговых данных, но прогнозная сила существенно падает на следующий временных участках. И здесь нам важна не столько стабильность популяции, сколько стабильность прогнозной значимости переменных.

Документ содержит пошаговое описание процесса построения в виде скрипта на языке R, комментарии к скриптам, объяснение процесса и логики принятия решений в ходе процесса подготовки данных и обучения модели, выводы при построении модели, а также итоговый инструмент — скоринговая карта и примеры таблиц распределений. В документе представлены все три модели.

Скоринговая модель может быть внедрена в системе принятия решений как аддитивная бальная модель. Скоринговая карта является удобным инструментом для интерпретации результатов оценки рисков и решения, анализа профиля клиента. Бизнес внедрение модели заключается в установлении баллов отсечения согласно стратегиям, например, достижение0020установленного уровня одобрения заявок, снижение уровня дефолта до заданного уровня. Данная модель дает множество вариантов движения вдоль стратегической кривой, показывающей соотношение между уровнем одобрения заявок и прогнозным уровнем дефолта портфеля. Данные соотношения зависят как от качества модели, так и в большей степени распределения популяции потока заявок по скоринговому баллу. Следует помнить, что какой бы точной ни была модель, в случае, когда большинство клиентов имеют профайл высокого риска, решение выдавать с низким уровнем отказа приведет к высоком дефолту. Например, если в потоке клиентов 80% некредитоспособных клиентов, решение одобрять 21% заявок приведет к минимум 1% дефолта при идеальной модели. При выборке стратегии мы можем двигаться только по стратегической кривой. Скоринг является инструментом оценки рисков, но не гарантирует достижение заданных бизнес-показателей, которые в кредитовании зависят во многом как от качества потока клиентов и процессов сопровождения кредита и взыскания, так и от грамотно внедренных финансовых и операционных моделей.