МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

Институт Цифровой экономики и информационных технологий

Кафедра Математических методов в экономике

«Допустить к защите»
Заведующий кафедрой
математических методов в экономике
_____Тихомиров Н.П.
«_____»_____2019 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Направление 01.03.02 «Прикладная математика и информатика»

Профиль «Прикладная математика и информатика»

Тема: «Оценка дефолтности заемщиков кредитных организаций»

Выполнил студент Эль-Айясс Дани Валид

Группа 444

Научный руководитель выпускной
квалификационной работы
Тихомиров Н.П., профессор, д.э.н.

_____

Автор _____

Москва − 2019

Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

АННОТАЦИЯ

выпускной квалификационной работы

Эль-Айясса Дани Валида

на тему: «Оценка дефолтности заемщиков кредитных организаций»

Данная выпускная квалификационная работа посвящена проблеме оценки кредитоспособности заемщиков кредитных организаций и построения скоринговой модели на основе накопленных кредитными организациями данных. В рамках поставленной проблемы проанализированы различные подходы и алгоритмы для построения скоринговой модели. Также продемонстрированы основные подходы к оценке качества построенной модели. Изложены общие сложности, которые возникают при решении подобных задач, а также особенности каждого подхода к их решению. Показаны основные проблемы при работе с реальными данными и методы их решения, принципы работы с разными типами данных. Обоснованы основные факторы, влияющие на кредитоспособность заемщика, и предложен лучший подход в рамках решения конкретной задачи. В заключении проведен сопоставительный анализ всех методов и их результатов на основе метрик качества. На основе лучшей модели были получены оценки вероятности дефолта для новых клиентов, которые могут быть использованы кредитными организациями для формирования нового кредитного портфеля в зависимости от политики кредитной организации и степени склонности к риску и других факторов.

Ключевые слова: кредитоспособность, дефолт, прогнозирование, машинное обучение, алгоритм, классификация.

ABSTRACT

«Evaluation of the default of borrowers of credit organizations»

This diploma thesis is dedicated to the problem of evaluating creditworthiness of borrowers of credit organizations and making scoring model using accumulated by credit organizations data. Within the framework of the problem, different approaches and algorithms for making scoring models and different approaches to the assessment of the quality of the constructed models was suggested. A set of approaches to solve general problems while solving similar and also features of each approach, its advantages and disadvantages was proposed. It also shows the main problems when working with real data and methods for their solution, principles of working with different types of data. In conclusion, a comparative analysis of all methods and their results based on quality metrics was carried out, the main factors affecting the borrower's creditworthiness were presented, and the best approach was proposed within the framework of solving a current task. Also, based on the best model, estimates of the probability of default for new customers were obtained, which can be used by credit organizations to form a new loan portfolio depending on the policy of the credit organization and the degree of risk appetite and other factors.

Keywords: creditworthiness, default, forecasting, machine learning, algorithm, classification.

Автор ВКР _____ Эль-Айясс Дани Валид

Содержание

Введение

В современном мире кредитные организации, в частности банки, предоставляют огромный спектр финансовых услуг, таких как расчетно-кассовое обслуживание, валютные операции, инвестиционные услуги, кредитование и др. Результатом оказания этих услуг является большое количество накопленных данных. Хранение такого количество информации весьма затратно. Но банки могут использовать эту информацию для повышения своей эффективности, качества обслуживания клиентов, а также управления рисками.

Кредитование физических и юридических лиц является одним из основных источников доходов банка, поэтому грамотная организация кредитного процесса является актуальной задачей, так как помогает избежать возникновения многих рисков и обеспечить максимальную прибыль от выданных в кредит денежных средств.

Для этого строятся *скоринговые модели*, задачей которых является прогнозирования кредитоспособности новых клиентов на основании данных о выданных ранее кредитах. Для построения таких моделей все чаще прибегают к различным методам *машинного обучения*.

В данной работе рассмотрен процесс построения моделей кредитного скоринга с использованием данных о заемщиках Home Credit Bank’a от предобработки данных до прогнозирования кредитоспособности новых клиентов.

Цель данной работы – построение скоринговой модели с помощью современных методов машинного обучения и формирование оптимального решения о выдаче кредита.

Для достижения поставленной цели необходимо решить следующие **задачи**:

1. проанализировать общие проблемы, возникающие при решении задач кредитного скоринга и способы их решения;

2. изучить различные алгоритмы для построения скоринговой модели, их особенности и недостатки;

3. рассмотреть основные подходы к оценке качества моделей;

4. изучить основные подходы работы с большими данными;

5. провести сопоставительный анализ всех методов и их результатов на основе метрик качества и изучить основные факторы, влияющие на кредитоспособность заемщика;

6. обосновать выбор лучшего метода в рамках решения задачи кредитного скоринга и на его основе получить оценки вероятности дефолта для новых клиентов.

Объектом исследования является дефолтность заемщиков банка Home Credit Bank.

Предметом исследования является количественная оценка заемщиков банка Home Credit Bank.

Методологическим инструментарием исследования выступают алгоритмы прогнозирования кредитоспособности по известным характеристикам заемщика на основе методов машинного обучения.

Данная работа состоит из введения, трех глав и заключения.

Во введении обоснована актуальность темы исследования, определены цель и задачи исследования, выделены объект и предмет исследования.

В первой главе раскрываются понятия кредита и кредитного скоринга, а также структура исходных данных.

Во второй главе рассмотрены основные понятия машинного обучения и типы моделей, которые использованы для оценки кредитоспособности заемщиков.

В третьей главе описаны результаты по каждой модели и произведен сопоставительный анализ.

В заключении подводятся итоги исследования, делаются основные выводы.

Глава 1. Теоретические основы оценки кредитоспособности заемщиков

1.1. Задача кредитного скоринга

Кредитная организация в соответствии с законодательством Российской Федерации – это юридическое лицо, которое для извлечения прибыли как основной цели своей деятельности на основании специального разрешения (лицензии) Центрального банка Российской Федерации (Банка России) имеет право осуществлять банковские операции, предусмотренные Федеральным законом «О банках и банковской деятельности» [1].

Виды кредитных организаций (в соответствии с законом РФ «О банках и банковской деятельности»):

1. *Банк* – кредитная организация, которая имеет исключительное право осуществлять банковские операции, такие, как: привлечение во вклады денежных средств физических и юридических лиц, размещение указанных средств от своего имени и за свой счёт на условиях возвратности, платности, срочности, открытие и ведение банковских счетов физических и юридических лиц.

2. *Небанковская кредитная организация* – кредитная организация, имеющая право осуществлять отдельные банковские операции, предусмотренные ФЗ «О банках и банковской деятельности». Допустимые сочетания банковских операций для небанковских кредитных организаций устанавливаются Банком России.

Основной цель кредитной организации является извлечение прибыли. Кредитование физических и юридических лиц является одним из основных источников доходов банка [6].

Кредит – это ссуда, предоставленная кредитной организацией заемщику под определенные проценты за пользование деньгами. Кредиты выдаются физическим и юридическим лицам.

Грамотная организация кредитного процесса помогает избежать возникновения многих рисков и обеспечить максимальную прибыль от выданных в кредит денежных средств [5].

Стоит дать определение терминам кредитоспособность и дефолт.

Кредитоспособность заемщика кредитной организации – это способность заемщика полностью рассчитаться долговым обязательствам. Степень риска выдачи кредита конкретному заемщику связана с уровнем его кредитоспособности [6].

Под *дефолтом* понимают невыполнение договора займа. Дефолт бывает двух видов:

1. банкротство заемщика, то есть невозможность выполнения заемщиком своих обязательств;
2. технический дефолт – ситуация, когда заёмщик нарушил договор займа, но физически он этот договор выполнять может.

Для кредитных организаций важно, чтобы деньги, выданные заемщику, были возвращены в полном размере вместе с процентом по кредиту, поэтому банк хочет минимизировать количество дефолтов заемщиков, поэтому кредиты должны быть выданы наиболее кредитоспособным заемщикам [18].

Кредитный скоринг – это задача, в которой прогнозируется риск дефолта заёмщика при его обращении в банк на получение кредита по данным его анкеты-заявления. Скоринг является одним из этапов проверки заемщика и его целью является минимизация риска на основе накопленных банком данных [19].

В задаче кредитного скоринга банку нужно принять решение о том, выдавать данному физическому или юридическому лицу кредит или нет. Особенность данной задачи в том, что здесь кроме бинарного решения *«плохой»* или *«хороший»*, *bad* или *good*[1], нужно также оценивать вероятность того, что будет дефолт: вероятность того, что данный заемщик окажется «плохим» [31].

[1] Термины bad и good используются кредитными аналитиками во всем мире – это сложившаяся и устоявшаяся терминология.

Стоит также заметить, что, согласно положению Банка России, финансовое положение заемщика может быть оценено как хорошее, среднее или плохое [2]. При этом нет четких определений данной градации.

На основе этой вероятности клиенты сортируются и банк получает упорядоченный список заемщиков [14]. Чем выше заемщик в списке, тем больше шансов, что он вернет кредит. В зависимости от политики, которой придерживается банк, и склонности к риску он выбирает порог отсечения. Чем больше банк склонен к риске, тем большему количеству заемщиков он выдаст кредит. В зависимости от ситуации на рынке, экономической и политической ситуации в стране и других причин, данный порог отсечения может пересматриваться многократно.

Одной из задач кредитного скоринга является автоматизации процесса принятия решения о выдаче заемщику кредита. До использования моделей данное решение принималось кредитным экспертом на основе его опыта и руководствуясь характеристиками клиента.

1.2. Факторы кредитного скоринга

Обычно, в качестве характеристик заемщика, выступают данные из анкет, которые заполняются при подаче заявки на кредит. Признаки характеризуют как экономические характеристики данного заемщика, так и социально-демографические.

Для анализа заемщиков, которые брали кредит ранее или имеют его в настоящее время могут также использоваться данные о кредитной истории из кредитного бюро. Информация о кредитной истории обычно берется за 1–2 года, так как при большем сроке могут произойти социально-экономические изменения, в результате которых характеристики новых клиентов будут выделяться на фоне старых, а при меньшем сроке есть возможность недооценить вероятность дефолта [14].

Переходя к объекту исследования, а именно к данным о заемщиках банка Home Credit Bank – данные представлены в виде таблицы, строкам которой

соответствует отдельный заемщик, а столбцам описание этого заемщика. Данные взяты с сайта Kaggle[2] [32].

Таблица состоит из следующих признаков (признаки расположены в следующем порядке: количественные, категориальные и бинарные):

1. AMT_INCOME_TOTAL – доход клиента в долларах в год;
2. AMT_CREDIT – сумма по кредиту;
3. AMT_ANNUITY – кредитный аннуитет;
4. AMT_GOODS_PRICE − стоимость товара, на который предоставляется кредит;
5. CNT_CHILDREN – количество детей;
6. CNT_FAM_MEMBERS – количество членов семьи;
7. DAYS_BIRTH − возраст клиента в днях на момент подачи;
8. DAYS_EMPLOYED − за сколько дней до подачи заявления клиент начал текущую работу;
9. DAYS_REGISTRATION − за сколько дней до подачи заявления клиент изменил свою регистрацию;
10. DAYS_ID_PUBLISH − за сколько дней до подачи заявления клиент изменил документ, удостоверяющий личность, с которым он подал заявку на кредит;
11. NAME_CONTRACT_TYPE – тип кредита, принимает 2 значения:
    - cash loans – обычный кредит;
    - revolving loans – возобновляемый кредит[3];
12. CODE_GENDER – пол заемщика, принимает 3 значения:
    - M – мужчина;
    - F – женщина;

[2] Kaggle – одна из наиболее известных платформ для проведения соревнований по Data Science, В каждом соревновании организаторы (в частности банки) выкладывают описание задачи, данные для решения этой задачи, метрику, по которой будет оцениваться решение и устанавливают сроки и призы.

[3] Возобновляемый кредит (револьверный кредит) – автоматически возобновляемая ссуда, обычно технически вводится в действие в виде овердрафта по кредитной карточке, в некоторых странах – в виде «диспо-кредита», то есть в виде права уходить в минус по обычному расчётному счёту.

- XNA – иное;

13. NAME_TYPE_SUITE − кто сопровождал клиента, когда он подал заявку на, принимает 7 значений:
    - unaccompanied – никто не сопровождал;
    - family – семья;
    - spouse, partner – супруг / супруга / партнер;
    - children – ребенок / дети;
    - group of people – группа людей;
    - other_A – другое А;
    - other_B – другое Б;
14. NAME_INCOME_TYPE – тип дохода клиента, принимает 8 значений:
    - working – работает;
    - commercial associate – коммерческий партнер;
    - pensioner – пенсионер;
    - state servant – государственный служащий;
    - unemployed – не работает;
    - student – студент;
    - businessman – бизнесмен;
    - maternity leave – декретный отпуск;
15. NAME_EDUCATION_TYPE – уровень образования, принимает 5 значений:
    - secondary / secondary special – среднее / среднее специальное;
    - higher education – высшее образование;
    - incomplete higher – неоконченное высшее;
    - lower secondary – неполное среднее;
    - academic degree – ученая степень;
16. NAME_FAMILY_STATUS – семейный статус клиента, принимает 6 значений:
    - married – состоит в браке;

- single / not married – не состоит в браке / одинок;
- civil marriage – гражданский брак;
- separated – разведен / разведена;
- widow – вдовец / вдова;

17. FLAG_OWN_CAR – наличие собственного автомобиля;
18. FLAG_OWN_REALTY – наличие собственного дома или квартиры;
19. FLAG_MOBIL − предоставил ли клиент мобильный телефон;
20. FLAG_CONT_MOBILE – был ли мобильный телефон доступен;
21. FLAG_EMP_PHONE – предоставил ли клиент рабочий телефон;
22. FLAG_EMAIL – предоставил ли клиент электронную почту.

Так как признаковое описание состоит из 22 признаков, среди которых 10 количественных признаков, 6 категориальных признаков и 6 бинарных признаков, то задача классификации клиентов решается в пространстве разнотипных признаков.

Глава 2. Методы машинного обучения в задаче кредитного скоринга

2.1. Теория машинного обучения

Машинное обучение, по большей части – это наука о том, как решать задачу восстановления функции по точкам (задача обучения с учителем).

2.1.1. Постановка задачи машинного обучения

Пусть имеется неизвестная функция отображения из множества объектов в множество ответов, но эта функция измерена только в конечном множестве точек. То есть имеется n штук пар объект-ответ, и по этой информации необходимо восстановить эту зависимость или построить функцию, аппроксимирующую эту неизвестную зависимость [7].

Введем обозначения:

- X – множество объектов;
- Y – множество ответов.
- $y : X \to Y$ – неизвестная зависимость.

Дано:

- $\{x_1, \dots, x_n\} \subset X$ – обучающая выборка;
- $y_i = y(x_i), i = 1, \dots, n$ – известные ответы.

Найти:

- $a : X \to Y$ – *алгоритм*, решающая функция, приближающая y на всем множестве X.

Самым распространённым способом задания объектов является признаковое описание. *Признаки* – это функции, ставящие объектам значения $f_j: X \to D_j$ [13], где

$D_j = \{0, 1\}$ – *бинарный* признак: ответ «да» или «нет» про интересующий объект;

$D_j < \infty$, D_j неупорядоченно[4] – *номинальный* или категориальный признак, принимающий конечное множество значений, больше 2;

$D_j < \infty$, D_j упорядоченно[5] – *порядковый* признак на множестве значений признаков задано некое отношение порядка;

$D_j = \mathbb{R}$ – *количественный* признак.

Обычно в прикладных задачах все эти признаки смешаны. Каждый тип признаков нужно по-разному предобрабатывать и по-разному учитывать в алгоритмах машинного обучения [13].

Главный объект, с которым постоянно приходится иметь дело – это *матрица объекты-признаки*, представление обучающей выборки:

$$\boldsymbol{F} = \left\| \boldsymbol{f_j}(\boldsymbol{x_i}) \right\|_{\boldsymbol{n \times m}} = \begin{pmatrix} \boldsymbol{f_1}(\boldsymbol{x_1}) & \cdots & \boldsymbol{f_m}(\boldsymbol{x_1}) \\ \vdots & \ddots & \vdots \\ \boldsymbol{f_1}(\boldsymbol{x_n}) & \cdots & \boldsymbol{f_m}(\boldsymbol{x_n}) \end{pmatrix}. \quad (1)$$

Строки этой матрицы – объекты, а столбцы – признаки. Каждой строке соответствует некоторый правильный ответ.

Вектор $(f_1(x), \dots, f_m(x))$ – признаковое описание объекта x.

В зависимости от того, какие именно ответы должны возвращать алгоритм зависит то, с каким типом задачи приходится иметь дело. Иными словами, тип задачи определяется пространством ответов.

Задачи классификации:

$Y = \{-1, +1\}$ или $Y = \{0,1\}$ – двухэлементное множество классов соответствует классификация на 2 класса (бинарная классификация). Это задачи, в которых необходимо принять одно из двух решений. Задача кредитного скоринга относится к этому типу. $y = 0$ – клиент кредитоспособен, $y = 1$ – клиент некредитоспособен;

$Y = \{1, \dots, M\}$ – классификация на M непересекающихся классов;

$Y = \{0,1\}^M$ – классификация на M классов, которые могут пересекаться.

Задачи восстановления регрессии:

[4] Нельзя сравнивать на больше или меньше. Сравнение только на равенство.
[5] Можно сравнивать на больше или меньше, но нельзя мерить расстояние друг от друга.

$Y = \mathbb{R}$ – ответами являются действительные числа.

Задачи ранжирования[6]:

Y – конечное упорядоченное множество.

Предсказательная модель обычно выбирается из некоторого параметрического семейства функций $A = \{a(x) = g(x, \theta) \mid \theta \in \Theta\}$, в котором нашлась бы функция, которая хорошо аппроксимировала бы неизвестную зависимость.

$g : X \times \Theta \to \mathrm{Y}$ – фиксированная функция,

Θ – множество допустимых значений параметра θ.

Задачи машинного обучения делятся на две стадии решения, на два этапа. На первом этапе по выборке строится функция, которая будет предсказывать значения на новых объектах.

Метод обучения $\mu : (X \times \mathrm{Y})^n \to A$ по выборке $X^n = (x_i, y_i)_{i=1}^n$ строится алгоритм $a = \mu(X^n)$:

$$\begin{pmatrix} \boldsymbol{f_1(x_1)} & \cdots & \boldsymbol{f_m(x_1)} \\ \vdots & \ddots & \vdots \\ \boldsymbol{f_1(x_n)} & \cdots & \boldsymbol{f_m(x_n)} \end{pmatrix} \overset{\boldsymbol{y}}{\to} \begin{pmatrix} \boldsymbol{y_1} \\ \cdots \\ \boldsymbol{y_n} \end{pmatrix} \overset{\boldsymbol{\mu}}{\to} \boldsymbol{a}. \qquad (2)$$

Вторая стадия – это тестирование на выборке из новых объектов. Такая выборка называется тестовой или контрольной. Построенный алгоритм должен выдавать на этих объектах некие ответы – прогнозы.

Алгоритм a для новых объектов $x'_1,\ldots, x'_k$ выдает ответы $a(x'_i)$.

$$\begin{pmatrix} \boldsymbol{f_1(x'_1)} & \cdots & \boldsymbol{f_m(x'_1)} \\ \vdots & \ddots & \vdots \\ \boldsymbol{f_1(x'_k)} & \cdots & \boldsymbol{f_m(x'_k)} \end{pmatrix} \overset{\boldsymbol{a}}{\to} \begin{pmatrix} \boldsymbol{a(x'_1)} \\ \cdots \\ \boldsymbol{a(x'_k)} \end{pmatrix}. \qquad (3)$$

Первая стадия является самой сложная. Вторая – это просто применение построенной функции к новым объектам.

Обычно задачу обучения сводят к задаче оптимизации, но, чтобы это сделать, нужно разобраться с тем, как измеряется точность ответа алгоритма на одном отдельном объекте [13].

Для этого вводится понятие *функции потерь*:

[6] Задачи решаются поисковыми системами при выдаче поисковой выдачи.

$\mathcal{L}(a, x)$ – величина ошибки алгоритма $a \in A$ на объекте $x \in X$.

Для задачи классификации – это просто индикатор ошибки, то есть был ли ответ правильный или нет:

$\mathcal{L}(a, x) = [a(x) \neq y(x)]$ – индикатор ошибки.

Для задач регрессии обычно берут квадратичную функцию потерь, которая приводит к широко известному методу наименьших квадратов [16]:

$\mathcal{L}(a, x) = (a(x) - y(x))^2$ – квадратичная ошибка.

Если сложить эти потери по всем объектам и поделить на количество объектов, то получится функционал[7], который называется *эмпирическим риском*.

Эмпирический риск – функционал качества алгоритма a на X^n:

$$\boldsymbol{Q(a, X^n) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(a, x_i).} \quad (4)$$

Для модели $a(x, \theta)$, ищется параметр θ, который будет давать минимум эмпирического риска. То есть задача обучения состоит в подборе такого алгоритма, на котором достигается минимум функционала ошибки [12]:

$$\boldsymbol{\mu(X^n) = arg \min_{a \in A} Q(a, X^n).} \quad (5)$$

Для решения поставленной оптимизационной задачи применяются различные численные методы.

2.1.2. Проблема переобучения

Одной из основных проблем машинного обучения является проблема *переобучения*. Она заключается в том, что алгоритм «подгоняется» к заданным точкам, а не восстанавливает исходную зависимость, то есть в других точках пространства объектов, которые алгоритм не видел (на которых он не обучался), он выдает неправильные и плохие ответы, то есть не обладает обобщающей способностью [21].

У борьбы с переобучением есть одна проблема. Дело в том, что и хороший алгоритм, который достаточно полно обобщает информацию, и переобученный

[7] Функционал, так как принимает на вход функцию.

алгоритм будут иметь хорошее качество на обучающей выборке. Отличаются они только по качеству на новых данных. Хороший алгоритм хорошо работает на новых данных, в то время как переобученный – плохо. То есть нельзя понять переобучился ли алгоритм или нет только лишь по обучающей выборке. Нужны дополнительные данные для выявления эффекта переобучения.

Стоит также упомянуть, что есть проблема *недообучения*. Недообучение состоит в том, что алгоритм имеет плохое качество и на обучающей выборке, и на новых данных. При этом с недообучением бороться проще: нужно усложнять семейство алгоритмов, то есть брать более сложные алгоритмы.

Для измерения переобучения используют функционалы, измеряющие качество построенной модели данных, на которых эта модель не обучалась, которых она ранее не видела.

Самый простой способ – поделить выборку на обучающую и контрольную (отложенную). X^n – это обучение, X^k – это контроль. Исходно есть $L = n + k$ точек, которые разбили на 2 части. На одной происходит обучение модели, на второй проверка. Эмпирический риск на тестовых данных:

$$\boldsymbol{HO(\mu, X^n, X^k) = Q(\mu(X^n), X^k) \rightarrow min.} \qquad (6)$$

Если взять отложенную выборку слишком маленькой, то обучающая выборка будет репрезентативной, но контрольная выборка окажется слишком маленькой, чтобы надежно оценить качество, и оценка качества будет зашумленной. Если же взять отложенную выборку слишком большой, то оценка по ней будет надежной, но обучение будет слишком маленьким и качество может быть небольшим. Обычно берут обучающую и контрольную выборки в пропорции 70:30 или 80:20 соответственно [10].

Преимуществом данного подхода является обучение алгоритма всего один раз, но при этом результат сильно зависит от разбиения.

Чтобы решить эту проблему можно разбить все данные на обучение и тест N раз и применить предложенный ранее подход. В результате получится N показателей качества, которые можно усреднить и получить итоговую оценку. Но и у данного подхода есть недостаток: поскольку разбиения случайные, все

еще нет гарантий, что каждый объект хотя бы раз побывает в обучении. Нужен более системный подход. Таким подходом является *кросс-валидация*, в котором вся выборка делится на N блоков примерно одинакового размера, и каждый блок по очереди будет выступает в качестве тестового, в то время как остальные выступают в качестве обучающей выборки. После этого получается N показателей качества, которые усредняются и получается оценку качества по кросс-валидации.

$$\boldsymbol{CV(\mu, X^L) = \frac{1}{N}\sum_{i=1}^{N} Q(\mu(X_i^n), X_i^k) \rightarrow min.} \quad (7)$$

У кросс-валидации та же проблема с выбором количества блоков, что и у отложенной выборки. Обычно выборку делят на 3, 5 или 10 блоков в зависимости от объема обучающей выборки [10].

2.1.3. Метрики качества в задаче кредитного скоринга

После построение модели необходимо понять, насколько хорошо она работает. Для этого используют метрики качества. Так как задача кредитного скоринга является задачей бинарной классификации – далее будут рассмотрены только метрики классификации.

Самым простым вариантом является *доля правильных ответов*:

$$\boldsymbol{Accuracy = \frac{1}{n}\sum_{i=1}^{n} [a(x_i) = y_i]}. \quad (8)$$

Но эта метрика обладает рядом недостатков. В частности, с интерпретацией на несбалансированных выборках, когда количество объектов одного класса сильно большего другого (задача кредитного скоринга является примером такой задачи). Для плохого алгоритм, который для всех объектов возвращает мажоритарный класс, доля правильных ответов составит долю мажоритарного класса (не 0.5, как может показаться). Поэтому для устранения данного недостатка можно измерять не только долю правильных ответов, но и

базовую долю правильных ответов, равную максимальному качеству, которое можно достичь, если алгоритм возвращает константу на всех объектах:

$$\boldsymbol{BaseRate} = \boldsymbol{arg}\max_{\boldsymbol{y_0}\in\{\boldsymbol{0,1}\}}\frac{\boldsymbol{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}[\boldsymbol{y_0} = \boldsymbol{y_i}]. \quad (9)$$

Другой недостаток метрики (8) заключается в том, что эта метрика не учитывает разные цены ошибок. Рассмотрим, какие виды ошибок может допускать классификатор.

Каждый объект характеризуется двумя числами:

y – правильный ответ на нем;

$a(x)$ – ответ, который дает алгоритм.

В зависимости от сочетания этих двух чисел, можно разделить все объекты на четыре категории:

«верные срабатывания» или True Positive. К ним относятся те объекты, на которых правильный ответ 1, и алгоритм возвращает 1.

«ложное срабатывание» или False Positive (ошибка 1-го рода). К ним относятся те объекты, на которых правильный ответ 0, и алгоритм возвращает 1.

«ложный пропуск» или False Negative (ошибка 2-го рода). К ним относятся те объекты, на которых правильный ответ 1, и алгоритм возвращает 0.

«верный пропуск» или True Negative. К ним относятся те объекты, на которых правильный ответ 0, и алгоритм возвращает 0.

Все эти категории можно представить с помощью матрицы ошибок, представленной в таблице 1:

Таблица 1 – Общий вид матрицы ошибок в случае бинарной классификации

	$y = 1$	$y = 0$
$a(x) = 1$	True Positive (TP)	False Positive (FP)
$a(x) = 0$	False Negative (FN)	True Negative (TN)

Доля правильных ответов, выраженная через эти показатели, выглядит следующим образом:

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP + TN}}{\boldsymbol{TP + TN + FP + FN}}. \quad (10)$$

Разные типы ошибок могут иметь разную цену. В задаче кредитного скоринга, в зависимости от политики банка, может быть, что ложный пропуск (выдать кредит плохому заемщику) хуже, чем ложное срабатывание (не выдать кредит хорошему заемщику), или же наоборот. В таких ситуациях лучше измерять две другие метрики качества: *точность и полноту* [35].

Точность показывает, насколько можно доверять классификатору, если он выдает ответ $a(x) = 1$ (заемщик дефолтный), определяется как:

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FP}}. \quad (11)$$

Полнота показывает, как много объектов класса $y = 1$ (заемщик дефолтный) алгоритм находит:

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FN}}. \quad (12)$$

В зависимости от того, какие ошибки приемлемы, можно отдавать предпочтения одной из метрик. Чем выше точность, тем меньше ложных срабатываний. Чем выше полнота, тем меньше ложных пропусков.

Эти две метрики можно объединить с помощью гармонического среднего или *F-меры*:

$$\boldsymbol{F} = \frac{\mathbf{2}}{\frac{\mathbf{1}}{\boldsymbol{precision}} + \frac{\mathbf{1}}{\boldsymbol{recall}}} = \frac{\mathbf{2} * \boldsymbol{precision} * \boldsymbol{recall}}{\boldsymbol{precision} + \boldsymbol{recall}}. \quad (13)$$

Также есть расширенная версия F-меры с параметром β:

$$\boldsymbol{F_\beta} = (\mathbf{1} + \boldsymbol{\beta^2}) \frac{\boldsymbol{precision} * \boldsymbol{recall}}{\boldsymbol{\beta^2} * \boldsymbol{precision} + \boldsymbol{recall}}. \quad (14)$$

Она позволяет отдавать предпочтение точности или полноте варьируя параметр β.

Классификатор, как правило, работает в два этапа. На первом он вычисляет *оценку принадлежности* объекта к классу один. Дальше эта оценку сравнивается с порогом. Если она больше порога, то объект относится к классу $y = 1$. Если меньше, то к классу $y = 0$:

$$\boldsymbol{a(x)} = [\boldsymbol{b(x)} > \boldsymbol{t}], \quad (15)$$

где:

$b(x)$ – оценка принадлежности к классу 1;

t – порог классификации.

В задаче кредитного скоринга, как правило, строится алгоритм, который оценивает вероятность дефолта.

После оценивания алгоритма могут получиться небольшие значения метрик качества. Проблема может быть как в неправильно выбранном пороге t, так и в самом алгоритме $b(x)$. Перейдем к конкретным методам оценивания качества алгоритма.

Первый, основан на кривой точности и полноты или *PR-кривой*. Чтобы построить ее, объекты сортируются по возрастанию их оценки принадлежности к первому классу. Далее рассматриваются различные пороги классификации. Сначала ни один объект не относится к классу $y = 1$, затем только первый с максимальной оценкой, затем два и так далее:

$$\boldsymbol{b}\left(\boldsymbol{x}_{(1)}\right) \leq \cdots \leq \boldsymbol{b}\left(\boldsymbol{x}_{(n)}\right), \qquad \boldsymbol{t_i} = \boldsymbol{b}\left(\boldsymbol{x}_{(i)}\right). \tag{16}$$

Для каждого порога рассчитываются точность и полнота соответствующего алгоритма, и наносится эта точка в осях «точность» и «полнота». Соединив точки, получается кривая точность и полноты или PR-кривая. Данная кривая показывает всевозможные соотношения точности и полноты в зависимости от порога.

Чем больше площадь под кривой, тем лучше ведет себя алгоритм. Поэтому на практике часто используют такую метрику, как *площадью под PR-кривой или AUC-PRC* [35] (см рис. 1):

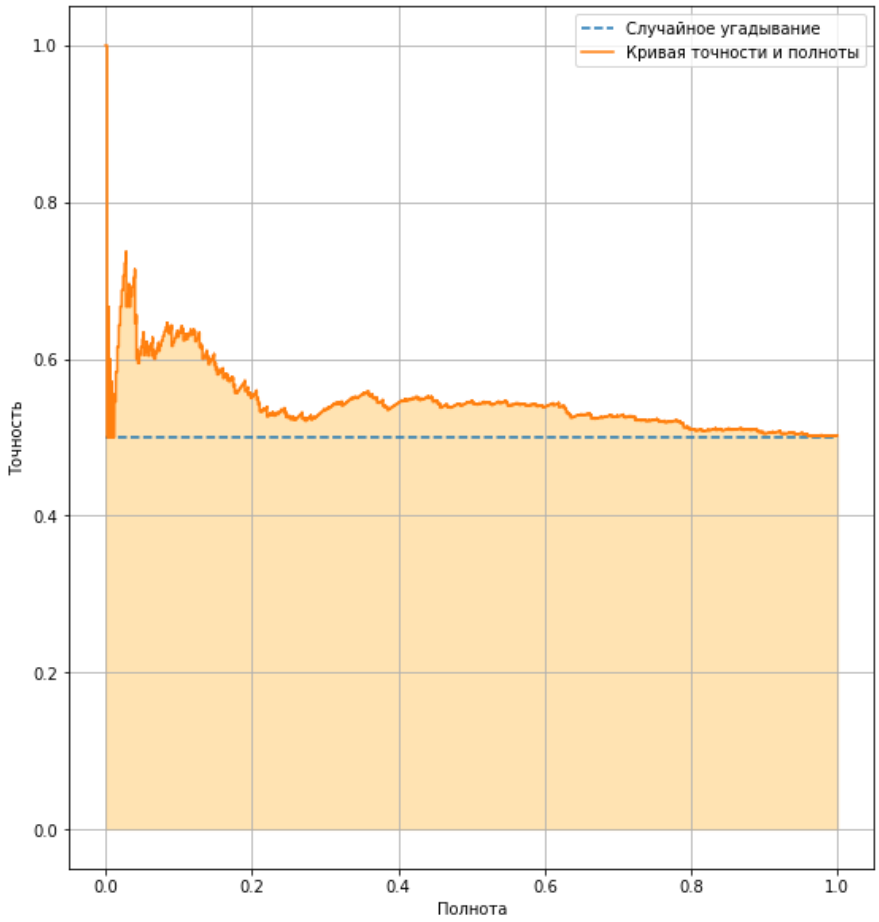

Рисунок 1 – Кривая точности и полноты и площадь под ней

Второй метод тоже основан на кривой, но в других координатах. Кривая называется *ROC-кривой*. По оси X отложена *доля объектов нулевого класса, ошибочно классифицируемых как объекты первого класса* (*False Positive Rate*):

$$\boldsymbol{FPR} = \frac{\boldsymbol{FP}}{\boldsymbol{FP} + \boldsymbol{TN}}. \quad (17)$$

По оси Y отложена *доля правильных положительных классификаций* (*True Positive Rate*):

$$\boldsymbol{TPR} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FN}}. \quad (18)$$

Данная кривая показывается всевозможные соотношения двух этих показателей в зависимости от порога.

Площадь под ROC-кривой тоже является хорошей метрикой качества, так и называется *AUC-ROC или площадь под ROC-кривой* (см рис. 2):

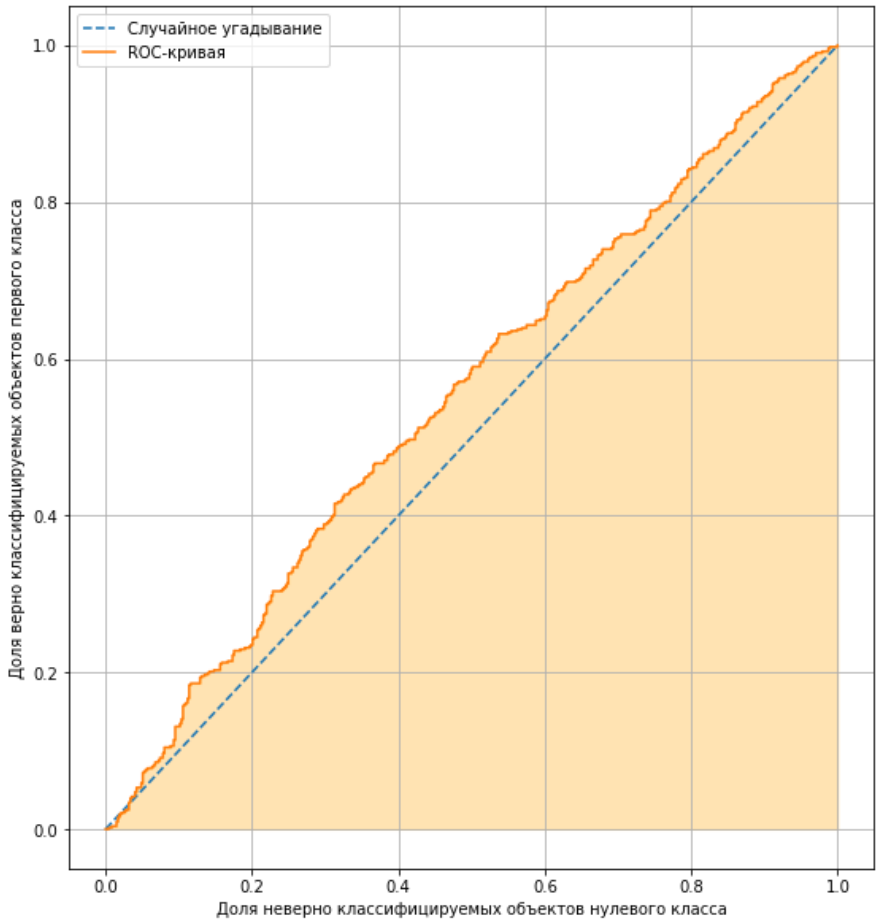

Рисунок 2 – ROC-кривая и площадь под ней

Данную метрику можно интерпретировать как долю пар объектов вида (объект класса $y = 1$, объект класса $y = 0$), которые алгоритм верно упорядочил, т.е. первый объект идёт в упорядоченном списке раньше. Или же можно сказать, что площадь под ROC-кривой равна вероятности того, что объект класса $y = 1$ получит вероятность больше, чем объект класса $y = 0$. В задаче же кредитного скоринга это формулируется как вероятность того, что дефолтный заемщик получит вероятность дефолтности больше, чем кредитоспособный заемщик. Исходя из этого данная метрика очень хорошо подходит для решения данной задачи.

2.1.4. Оптимизация гиперпараметров моделей

Отличие параметров модели от *гиперпараметров* заключается в том, что параметры модели изменяются и оптимизируются в процессе обучения модели и итоговые значения этих параметров являются результатом ее обучения, а

гиперпараметры модели задаются до начала ее обучения и не изменяется в процессе обучения. Эти параметры настраивают так, чтобы модель оптимизировала заданную функцию потерь на данных [10].

Существуют следующие основные подходы в подборе гиперпараметров:

- поиск по сетке;
- случайный поиск;
- байесовская оптимизация;
- оптимизация на основе градиентов.

Поиск по сетке заключается в полном переборе гиперпараметров по заданному подмножеству пространства обучающего алгоритма.

Так как пространство параметров алгоритма может включать вещественные или неограниченные параметры, то в таком случае производят дискретизацию и устанавливают границу для применения поиска по сетке. Поиск по сетке выдаёт в качестве результата лучший результат, достигнутый на процедуре проверки.

Плюсом этого подхода является возможность распараллеливания, так как алгоритмы можно запускать и обучать независимо друг от друга с разными гиперпараметрами.

Случайный поиск, исходя из названия, заключается не в полном переборе всех комбинаций, как в поиске по сетке, а в выборе этих комбинаций случайным образом [34].

Преимуществом случайного поиска, как и поиска по сетке является возможность распараллеливания. Так же для гиперпараметров можно указать распределения, то есть вместо дискретизации множества значений гиперпараметра можно просто задать равномерное или любое другое распределение этого гиперпараметра.

Байесовская оптимизация – итеративный метод глобальной оптимизации в пространстве гиперпараметров. В данном подходе выбор следующей точки учитывает результаты предыдущих. Идея состоит в том, чтобы на каждом шаге найти компромисс между исследованием точек в пространстве гиперпараметров

рядом с удачными найденными точками и исследованием точек с большой неопределенностью, где теоретически могут находиться более удачные точки [17].

Так как байесовская оптимизация является более направленным с точки зрения поиска оптимума подходом, то данный метод показывает лучшие результаты с меньшими вычислениями в сравнения с двумя предыдущими подходами. Данный метод является предпочтительным в случае, если проверка точки (обучение и проверка модели с заданными гиперпараметрами) осуществляется долго.

Оптимизация на основе градиентов. Для некоторых алгоритмов можно вычислить градиент гиперпараметров и оптимизировать их с помощью численных методов, например, методом градиентного спуска. То есть использовать метод градиентного спуска в пространстве гиперпараметров.

В данной работе подбор гиперпараметров будет осуществляться по сетке с помощью кросс-валидации.

2.2. Алгоритмы машинного обучения

Линейные модели. В задаче регрессии линейная модель представляет из себя взвешенную сумму всех признаков, то есть скалярное произведение вектора признаков x на вектор весов модели w. В задаче классификации линейная модель, это знак скалярного произведения вектора признаков x на вектор весов модели w:

$$\boldsymbol{a(x,w) = sgn\langle x,w\rangle}. \qquad (19)$$

Геометрически это означает, что в пространстве признаков строится гиперплоскость. В зависимости от знака скалярно произведения, объект находится по одну сторону от гиперплоскости. Если знак скалярного произведения положительный, то объект относим к первому класс, в противном случае к нулевому [28].

Перейдем к поиску параметров w. Если использовать самую простую функцию потерь, такую как индикатор ошибки (пороговая функция потерь), то

функционал эмпирического риска по вектору весов w неудобно минимизировать, так как он не является дифференцируемым. Для решения этой проблемы используют метод подмены функционала. Вместо индикатора ошибки используются его непрерывная аппроксимация. Для введения такой аппроксимации вводится понятие *отступа*:

$$\boldsymbol{M_i(w) = \langle x_i, w\rangle y_i}. \qquad (20)$$

Отступ оценивает, насколько далеко объект находится от разделяющей гиперплоскости, причем знак отступа показывает, насколько правильна классификация. Если знак положительный, значит ошибки нет, если знак отрицательный – ошибка есть, а абсолютная величина отступа показывает, насколько далеко объект находится от разделяющей гиперплоскости [36]. Поэтому отступ можно использовать для штрафа за то, что объект попал в другой класс, то есть отступ отрицательный. Чем больше по абсолютной величине отрицательный отступ, тем больше должен быть штраф. Следовательно, можно ввести непрерывную аппроксимацию с помощью некоторой невозрастающей непрерывной функцией от отступа:

$$\boldsymbol{\mathcal{L}(a, y) = [\langle x_i, w\rangle y_i < 0] \leq \mathcal{L}(\langle x_i, w\rangle y_i)}. \qquad (21)$$

Чем больше значение отступа, тем увереннее классификация и меньше значение функции потерь. Таким образом можно воспользоваться оценкой сверху для каждого слагаемого и получить новый, дифференцируемый по параметрам, функционал:

$$\boldsymbol{Q(w) = \frac{1}{n}\sum_{i=1}^{n}[a(x_i, w)y_i < 0] \leq \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(\langle x_i, w\rangle y_i) \to \min_w} \qquad (22)$$

Новый функционал мажорирует функционал эмпирического риска (число ошибок на обучающей выборке) сверху. Поэтому при минимизации нового функционала также будет происходить минимизация функционала числа ошибок. Разные функции потерь приводят к разным алгоритмам машинного обучения [21].

Приведем примеры функций потерь (см рис. 3):

- $[M < 0]$ – пороговая функция потерь;

- $max\ (0,1-M)$ – кусочно-линейная функция потерь (метод опорных векторов);
- $log_2(1+e^{-M})$ – логарифмическая функция потерь (логистическая регрессия);
- e^{-M} – экспоненциальная функция потерь (метод адаптивного бустинга);
- $(1-M)^2$ – квадратичная (линейный дискриминант Фишера).

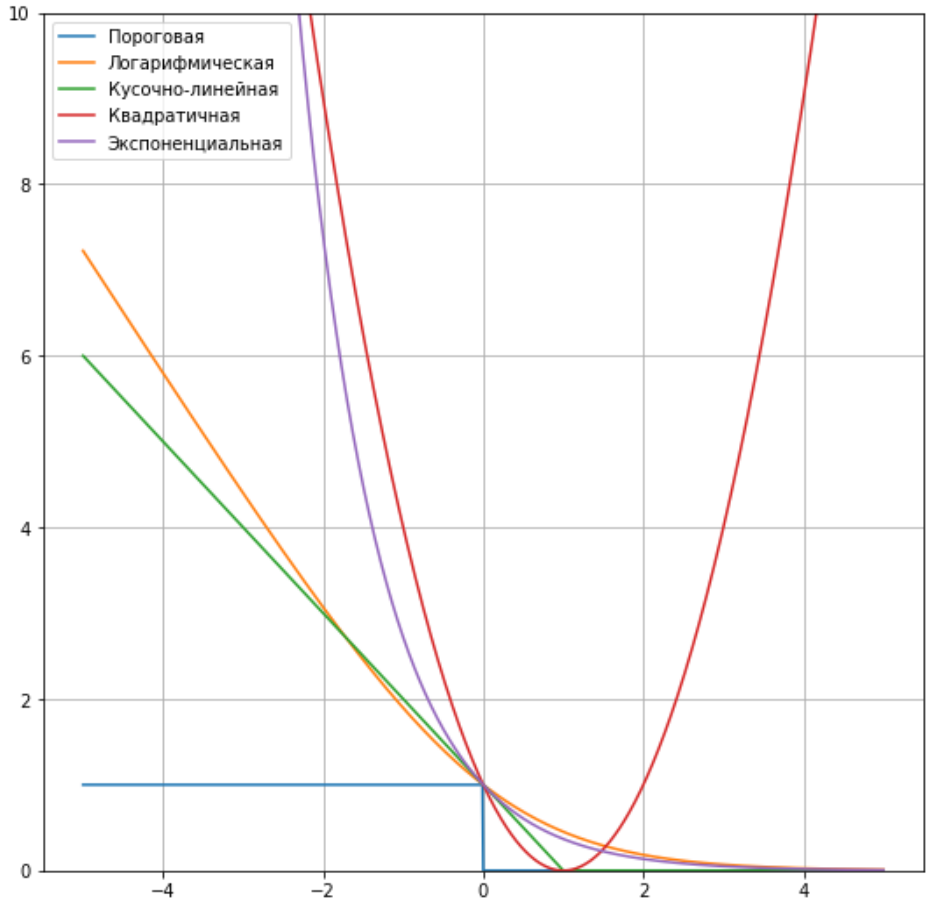

Рисунок 3 – Функции потерь

Логистическая регрессия. Предположим, что объекты обучающей выборки независимы и взяты из одного параметрического семейства распределений $(x_i, y_i)_{i=1}^{n} \sim p(x, y; w)$.

Для определения параметров w используют *метод максимального правдоподобия* [26]. В предположение, что плотность $p(x_i)$ не зависит от параметра модели w, а параметр модели используется в апостериорной

вероятности класса y_i для данного объекта x_i, представим совместную плотность распределения в следующем виде:

$$\boldsymbol{p(x_i, y_i; w) = P(y_i|x_i; w)p(x_i)}. \quad (23)$$

Тогда имеем следующее:

$$\boldsymbol{L(w) = \log \prod_{i=1}^{n} p(x_i, y_i; w) = \sum_{i=1}^{n} \log P(y_i|x_i; w)\, p(x_i) \to \max_{w}}. \quad (24)$$

Оказывается, что если предположить, что апостериорная вероятность имеет следующий вид:

$$\boldsymbol{P(y_i|x_i; w) = \frac{1}{1 + e^{-\langle x_i, w\rangle y_i}} = \sigma(\langle x_i, w\rangle y_i)}, \quad (25)$$

где $\sigma(M) = \frac{1}{1+e^{-M}}$ – *сигмоидная функция*, то принцип максимума правдоподобия дает ровно тот же функционал, который был введен раньше, то есть принцип максимума правдоподобия эквивалентен минимуму эмпирического риска с использованием логарифмической функции потерь:

$$\boldsymbol{Q(w) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-\langle x_i, w\rangle y_i}\right) \to \min_{w}}. \quad (26)$$

Данную задачу решают как методами первого порядка, например, градиентным спуском или его модификациями, так и методами второго порядка, таким как метод Ньютона-Рафсона [8].

Исходя из всего вышесказанного, логистическая регрессия позволяет оценить апостериорную вероятность класса для каждого классифицируемого объекта, что и требовалось получить в задаче кредитного скоринга [30].

Если $P(y = 1|X) > 0.5$, то объект принадлежит классу $y = 1$, в противном случае классу $y = 0$. Порог в 0.5 может быть изменен в зависимости от политика банка, как упоминалось ранее [24].

Из недостатков данного подхода можно выделить следующие:

- модель является линейной;
- чувствительность к корреляции между признаками – данная проблема может быть решена с помощью удаление сильнокоррелирующих

признаков, использованием метода главных компонент или же с помощью регуляризации;

- чувствительность к выбросам.

Метод опорных векторов. Имеем линейный классификатор $a(x, w, w_0) = sgn(\langle x, w\rangle - w_0)]$, где $\langle x, w\rangle - w_0$ – *дискриминантная функция*.

Сведем задачу к минимизации эмпирического риска для нахождения параметров. Отступом в данном случае будет являться произведением дискриминантной функции на метку правильного ответа. В качестве непрерывной аппроксимации пороговой функции потерь будет использована кусочно-линейная функция $max\ (0{,}1 - M)$. Минимизация эмпирического риска выглядит следующим образом:

$$\begin{aligned} \boldsymbol{Q(w, w_0)} &= \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}[(\langle \boldsymbol{x_i}, \boldsymbol{w}\rangle - \boldsymbol{w_0})\boldsymbol{y_i} < \mathbf{0}] \\ &\leq \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{max}\ (\mathbf{0}, \mathbf{1} - (\langle \boldsymbol{x_i}, \boldsymbol{w}\rangle - \boldsymbol{w_0})\boldsymbol{y_i}) \rightarrow \min_{\boldsymbol{w, w_0}}. \end{aligned} \tag{27}$$

Также введем *L2-регуляризатор*, который будет штрафовать решение за слишком большую Евклидову норму вектора весов. Данное введение помогает бороться с переобучением.

$$\begin{aligned} \boldsymbol{Q(w, w_0)} &= \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}[(\langle \boldsymbol{x_i}, \boldsymbol{w}\rangle - \boldsymbol{w_0})\boldsymbol{y_i} < \mathbf{0}] \\ &\leq \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{ma\,x}(\mathbf{0}, \mathbf{1} - (\langle \boldsymbol{x_i}, \boldsymbol{w}\rangle - \boldsymbol{w_0})\boldsymbol{y_i}) + \frac{\mathbf{1}}{\boldsymbol{2C}}\|\boldsymbol{w}\|^2 \\ &\rightarrow \min_{\boldsymbol{w, w_0}}. \end{aligned} \tag{28}$$

Задача, эквивалентная минимизации эмпирического риска, выглядит следующим образом:

$$\begin{cases} \dfrac{\mathbf{1}}{\boldsymbol{2C}}\|\boldsymbol{w}\|^{\mathbf{2}} + \displaystyle\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{\xi_i} \rightarrow \min\limits_{\boldsymbol{w, w_0, \xi}}, \\ \boldsymbol{M_i(w, w_0)} \geq \mathbf{1} - \boldsymbol{\xi_i}, \boldsymbol{i} = \mathbf{1}, \dots, \boldsymbol{n}, \\ \boldsymbol{\xi_i} \geq \mathbf{0}, \boldsymbol{i} = \mathbf{1}, \dots, \boldsymbol{n}. \end{cases} \tag{29}$$

Первый вариант задачи в виде единого функционала без дополнительных переменных ξ_i неудобна тем, что функционал в ней не является гладким. Поэтому принято решать задачу, которая представлена в виде квадратичного гладкого функционала с дополнительными переменными ξ_i, при которой функционал и ограничения являются гладкими. Данную задачу можно решить с помощью *метода Каруша-Куна-Таккера*. Исходя из математических условий, задача имеет единственное решение.

Преимущество метода в том, что ищется разделяющая гиперплоскость максимальной ширины. Это помотает делать уверенную классификацию. Недостатком метода является:

- чувствительность к выбросам и стандартизации данных;
- линейность.

Для решения проблемы линейности можно обобщить данный метод на случай нелинейной гиперплоскости с помощью использования ядер. Функция от пары объектов $K(x, x')$ называется *ядром*, если она представима в виде скалярного произведения $K(x, x') = \langle \varphi(x), \varphi(x') \rangle$ при некотором преобразовании $\varphi: X \to H$ из пространства признаков X в новое спрямляющее пространство H [29].

Деревья принятия решений или решающие деревья – важный класс алгоритмов машинного обучения. Принцип работы решающих деревьев очень схожа с тем, как люди принимают решения [9]. Построение дерева происходит следующим образом: объекты обучающей выборки последовательно разделяются на классы на основе самого значимого признака так, чтобы эти классы как можно сильнее отличались между собой с точки зрения целевой функции y.

Преимуществам метода следующие:

- быстрая построение;
- легкая интерпретируемость;
- отсутствие ограничения на некоррелируемость признаков;
- возможность работы с пропущенными значениями;

- нечувствительность к масштабированию;
- одинаково хорошо обрабатываются как непрерывные, так и категориальные признаки;
- восстановление нелинейных зависимостей.

К недостаткам метода можно отнести неоднозначность алгоритма построения структуры дерева, а также вопрос о том, когда стоит прекратить дальнейшее разделение на классы. Также деревья очень чувствительны к изменениям в выборке, то есть сильно меняются даже при небольшом изменении выборки. Из-за этого деревья слишком подгоняются под обучающую выборку (переобучаются). Бороться с переобучением довольно сложно. Надо либо использовать критерии остановок, которые слишком простые и не всегда помогают, либо делать стрижку деревьев, которая, наоборот, слишком сложная.

Черту решающих деревьев сильно меняться при небольшом изменении выборки можно использовать как преимущество при объединении деревьев в *композицию*, то есть для построения одного не переобученного алгоритма на основе большого количества решающих деревьев с лучшей обобщающей способностью [27]. Деревья решений можно использовать непосредственно для прогнозирования кредитоспособности заемщиков, но и использовать для объединения в композиции.

Для построения композиций деревьев используют два основных подхода:

- *беггинг* – подход, при котором алгоритмы обучаются независимо по случайным подвыборкам длины l с повторениями;
- *бустинг* – последовательное построение композиции алгоритмов, когда каждый следующий алгоритм стремится компенсировать ошибки и недостатки композиции всех предыдущих алгоритмов.

Случайный лес. Если совместить идею беггинга с методом случайных подпространств, суть которого заключается в обучении алгоритма на случайном подмножестве признаков, то это приводит к известному алгоритму под названием случайный лес. Результирующий ответ алгоритма выбирается с помощью простого голосования базовых деревьев. Данный алгоритм показывает

очень хорошее качество на практике. Преимущества данного метода такие же как у базового алгоритма – решающего дерева. Так же есть возможность независимого обучения базовых деревьев, то есть распараллеливание процесса обучения. К недостаткам можно отнести большой размер получающихся моделей.

Градиентный бустинг. Алгоритм, использующий идею бустинга, который строит линейную комбинацию базовых алгоритмов (чаще всего решающих деревьев) минимизирую функционал ошибки. На практике чаще всего превосходит все остальные алгоритмы по качеству. Преимущества бустинга следующие:

- возможность использования различных функций потерь;
- возможность использования любого семейства базовых алгоритмов.

Из недостатков метода хотелось бы отметить следующие:

- трудоемкость построения модели;
- сильная склонность к переобучению модели – подстраивается под данные;
- результаты бустинга сложно интерпретируемы.

Метрические алгоритмы и метод k ближайших соседей. Это семейство алгоритмов, которые используют функции *расстояния или метрики* в пространстве объектов. Исходная идея их использования лежит в предположении, что близким объектам соответствуют близкие ответы (гипотеза непрерывности) для задач регрессии и предположении о том, что близкие объекты лежат в одном классе (гипотеза компактности). Близость между объектами задается с помощью функции расстояния – функции от пары объектов, с неотрицательной областью значений. Иногда накладывают дополнительное требование, чтобы эта функция была метрикой [11].

Метод k ближайших соседей заключается в том, чтобы отнести объект u к тому классу, который доминирует (объектов этого класса больше, чем других) среди k ближайших объектов.

$$\boldsymbol{a(u; X^n, k) = arg\max_{y \in Y} \sum_{i=1}^{k} [y_u^{(i)} = y]}. \quad (30)$$

У такого подхода есть недостаток, заключающийся в том, что все объекты имеют одинаковый вес. Чтобы это исправить, можно ввести, весовую функцию w_i, которая оценивает важность i-го соседа. Тогда получается *метод k взвешенных соседей*:

$$\boldsymbol{a(u; X^n, k) = arg\max_{y \in Y} \sum_{i=1}^{k} [y_u^{(i)} = y] w_i}. \quad (31)$$

Например, в качестве весовой функции можно взять геометрическую прогрессию $w_i = q^i$, где $0 < q < 1$. Но в таком подходе никак не учитываются расстояния до объекта. Поэтому w_i можно определить как функцию от расстояния $\rho(u, x_u^{(i)})$, а не от ранга соседа как в предыдущем подходе. Введя дополнительно функцию ядра $K(z)$, невозрастающую на $[0, \infty)$, получается *метод парзеновского окна*:

$$\boldsymbol{a(u; X^n, k) = arg\max_{y \in Y} \sum_{i=1}^{k} [y_u^{(i)} = y] K\left(\frac{\rho\left(u, x_u^{(i)}\right)}{h}\right)}, \quad (32)$$

где h – это гиперпараметр, который называется *шириной окна*.

Наивный байесовский классификатор. Алгоритм, который основывается на *теореме Байеса*, но использует предположение, что наличие признаков в классе не связано с наличием другого признака. Из-за этого предположения алгоритм называется *«наивным»*.

Теорема Байеса выглядит следующим образом:

$$\boldsymbol{P(y|X) = \frac{P(X|y)P(y)}{P(X)}}, \quad (33)$$

где:

$P(y|X)$ – апостериорная вероятность класса y при данных значениях признаков X;

$P(X|y)$ – правдоподобие, вероятность данных значений X признаков при условии класса y;

$P(y)$ – априорная вероятность класса y;

$P(X)$ – априорная вероятность данных значений признаков X.

Теорема Байеса позволяется рассчитать *апостериорную вероятность* на основе *априорной* [17].

Так как используется «наивное» предположение о независимости признаков, то данную формулу можно преобразовать следующим образом:

$$\boldsymbol{P(y|X) = \frac{P(X|y)P(y)}{P(X)} = \frac{P(x_1|y)P(x_2|y) \dots P(x_m|y)P(y)}{P(X)}}, \quad (34)$$

где $P(y|x_i)$ – апостериорная вероятность класса y, при значении признака x_i.

Для выбора класса используется *максимум апостериорной вероятности*. Так как $P(X)$ является константой, то можно использовать следующее решающее правило:

$$\boldsymbol{\hat{y} = arg\max_y P(x_1|y)P(x_2|y) \dots P(x_n|y)P(y)}. \quad (35)$$

Для качественных признаков вероятности рассчитываются как частоты появления, а для количественных делается предположение о распределении (обычно используют нормальное распределение):

$$\boldsymbol{P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}}}. \quad (36)$$

Преимуществом алгоритма является простота и быстрота реализации, а также возможность работы с пропущенными данными. Недостатком модели является то самое «наивное» предложение о независимости признаков.

Алгоритмы машинного обучения являются разнообразными по своим постановкам и подходам к решению задач. Каждый подход имеет как свои преимущества, так и недостатки и нет универсальной модели.

В следующей главе будут построены все описанные выше модели, и произведен их сопоставительный анализ на основе описанных ранее метрик качества.

Глава 3. Разработка скоринговой модели

3.1. Подготовка исходных данных кредитоспособности заемщиков

Перед построением модели предварительно необходимо провести *предобработку данных*, так как в реальных данных приходится сталкиваться с рядом проблем [4]. В таблице 2 представлены характеристики первых пять объектов обучающей выборки (всего имеется 307511 объектов):

Таблица 2 – Характеристики первых пяти объектов обучающей выборки

Номер клиента	Дефолт/не дефолт	Тип кредита	Пол	Наличие собственного автомобиля	Наличие собственного дома или квартиры	Кол-во детей	Доход
1	2	3	4	5	6	7	8
100002	1	Кредит	M	N	Y	0	202500
100003	0	Кредит	F	N	N	0	270000
100004	0	Возобновляемый кредит	M	Y	Y	0	67500
100006	0	Кредит	F	N	Y	0	135000
100007	0	Кредит	M	N	Y	0	121500

Продолжение таблицы 2

Сумма по кредиту	Кред. аннуитет	Стоимость товара, на который дан кредит	Сопровождающий клиента при подаче заявления на кредит	Тип дохода	Семейный статус	Возраст	Кол-во дней работы на последнем месте
9	10	11	12	13	14	15	16
406597	24700	351000	Никто	Работает	Один	-9461	-637
1293502	35698	1129500	Семья	Гос. служащий	В браке	-16765	-1188
135000	6750	135000	Никто	Работает	Один	-19046	-225
312682	29686	297000	Никто	Работает	Один	-19005	-3039
513000	21865	513000	Никто	Работает	Один	-19932	-3038

Продолжение таблицы 2

За сколько дней до подачи заявления клиент изменил свою регистрацию	За сколько дней до подачи заявления клиент изменил документ, удостоверяющий личность, с которым он подал заявку на кредит	Дал ли клиент рабочий телефон	Дал ли клиент мобильный телефон	Был ли мобильный телефон доступен	Дал ли клиент эл. почту	Кол-во членов семьи
17	**18**	**19**	**20**	**21**	**22**	**23**
-3648	-2120	1	1	1	0	1
-1186	-291	1	1	1	0	2
-4260	-2531	1	1	1	0	1
-9833	-2437	1	1	1	0	2
-4311	-3458	1	1	1	0	1

Как мы можем увидеть, есть признаки, которые представляют из себя *текстовые значения*. Алгоритм не сможет понять, что эта за величина, поэтому используют следующий подход: преобразуют все уникальные значение в рамках каждого такого признака в числа от 0 до количества уникальных значений минус 1. Для новых клиентов будет использоваться то же самое преобразование с тем же самым отображением [22].

В реальных задачах часто приходится сталкиваться с *пропущенными значениями* в наблюдениях, которые могут носить как случайный, так и неслучайный характер. Данная проблема усложняет процесс построения и влияет на ухудшение качества. Если обучающая выборка достаточно большая, а наблюдений с пропусками мало, то их можно просто убрать из выборки, в противном случае прибегают к методам восстановления пропусков [20].

В данном наборе данных имеется 4 признака, содержащих пропущенные значения (см табл. 3):

Таблица 3 – Перечень признаков, имеющих пропущенные значения

Признак	**Количество пропусков**
Кредитный аннуитет	12
Стоимость товара, на который дан кредит	278
Сопровождающий клиента при подаче заявления на кредит	1292
Количество членов семьи	2
Всего объектов с 1 и более пропусков	1304

Так как выборка очень большая (более 300 тысяч наблюдений), то потеря 1304 наблюдений не является критичной.

Перейдем к анализу количественных данных. Такие признаки, как *возраст*, *количество дней работы на последнем месте*, *за сколько дней до подачи заявления клиент изменил свою регистрацию*, *за сколько дней до подачи заявления клиент изменил документ, удостоверяющий личность, с которым он подал заявку на кредит*, представлены как отрицательные значения, поэтому их необходимо домножить на -1.

Перейдем к *корреляционному анализу*. Рассмотрим визуализацию корреляционной матрицы (см. рис 4):

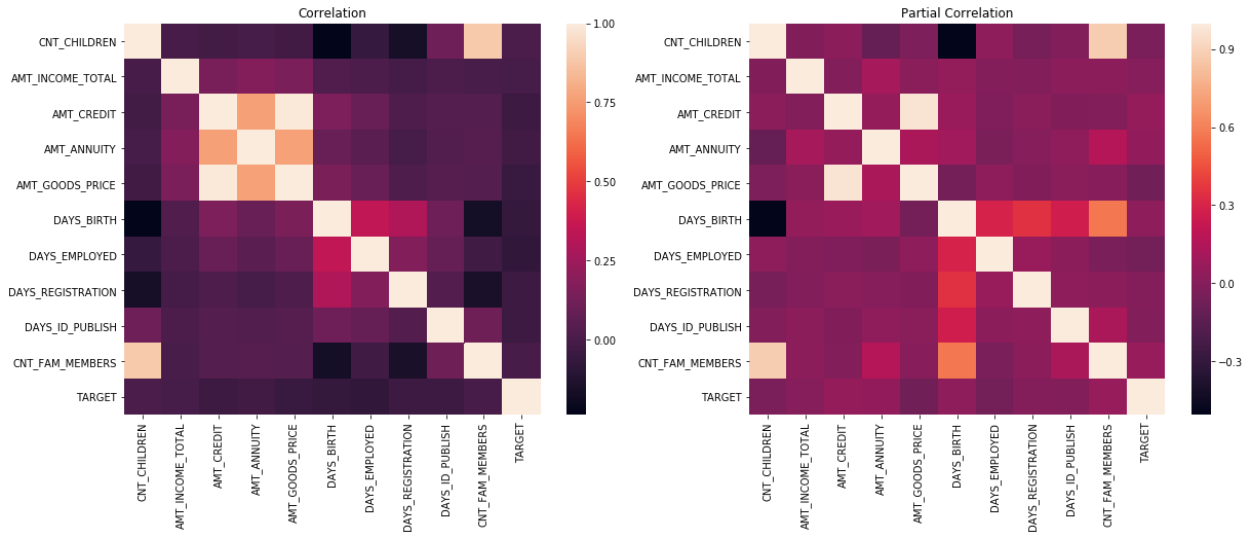

Рисунок 4 – Визуализация корреляционной матрицы

Все коэффициенты корреляции являются значимыми.

Есть признаки, корреляция между которыми очень высока. В таблице 4 приведены такие признаки:

Таблица 4 – Коррелирующие признаки

Признак 1	**Признак 2**	**Значение коэффициента парной корреляции**
Сумма по кредиту	Стоимость товара, на который дан кредит	0.9864
Количество детей	Количество членов семьи	0.8933
Кредитный аннуитет	Стоимость товара, на который дан кредит	0.7657
Кредитный аннуитет	Сумма по кредиту	0.7610

Для моделей, чувствительных к корреляции между признаками, необходимо убрать коррелирующие признаки и оставить один [3]. Исходя из таблицы выкинем следующие признаки: *стоимость товара, на который дан кредит, количество членов семьи, кредитный аннуитет*.

Далее, для каждого объекта выборки было рассчитано *расстояние Махаланобиса* по всем признакам до всех остальных объектов:

Таблица 5 – Пять объектов с самыми большими расстояниями Махаланобиса

Номер клиента	**Расстояние Махаланобиса**	**Значение F-статистики**	**P-значение**
114967	207268.92	25907.79	1.11e-16
336147	4825.64	603.18	1.11e-16
385674	2673.21	334.14	1.11e-16
190160	1175.39	146.91	1.11e-16
252084	657.55	82.19	1.11e-16

Для обучения моделей, чувствительных к выбросам, таких как, например, логистическая регрессия и метод опорных векторов, устраним из выборки объекты, расстояния от которых до всех остальных слишком велико [15][25].

Также для моделей, которые не связаны с решающими деревьями будут произведены такие подходы, как *стандартизация признаков* и *нормализация*

данных на отрезок [0, 1] (минимальному значению будет соответствовать 0, а максимальному 1) [33].

Перейдем к анализу категориальных и бинарных данных. Для каждого такого признака была построена таблица сопряженности и рассчитана статистика χ^2 для проверки гипотезы независимости наблюдаемых частот в таблице сопряженности [25] (см табл. 6):

Таблица 6 – χ^2 статистики для таблиц сопряженности категориальных признаков

Признак	**Статистика χ^2**	**P-значение**
Тип дохода	1256.27	4.76e-267
Пол	925.97	2.21e-203
Предоставил ли клиент рабочий телефон	650.94	1.39e-143
Семейный статус	507.47	1.61e-108
Тип кредита	297.09	1.41e-66
Наличие собственного автомобиля	145.13	2.00e-33
Сопровождающий клиента при подаче заявления на кредит	32.82	1.13e-05
Наличие собственного дома или квартиры	13.64	0.0002
Предоставил ли клиент мобильный телефон	2.36	0.12
Предоставил ли клиент электронную почту	0.86	0.35
Был ли мобильный телефон доступен	0.019	0.88

Как можно понять по таблице 6, гипотеза H_0 не отвергается для трех признаков.

Для таких моделей, как случайный лес, градиентный бустинг и наивный байесовский классификатор имеет смысл использовать категориальные переменные в их исходном виде [23]. Но несмотря на это будет использовано *частотное кодирование признаков,* суть которого заключается в замене каждой категории на частоту появления данной категории. Такое преобразование признака на практике помогает улучшить качество работы алгоритма [33].

Большое значение при построении моделей имеет соотношение классов, то есть соотношение кредитоспособных и некредитоспособных заемщиков. Сильное превышение одного из классов ведет к увеличению доли предсказаний в пользу мажоритарного класса.

При этом разные алгоритмы по-разному чувствительны как дисбалансу классов. Для логистической регрессии это может привести к тому, что алгоритм будет выдавать всем наблюдениям мажоритарный класс. Для борьбы с таким эффектом существует два подхода:

- удаление объектов мажоритарного класса;
- увеличение числа объектов миноритарного класса.

Удаление, так и увеличение объектов может происходить как случайным образом, так и другими специальными методами [21].

В данной задаче соотношение дефолтных и не дефолтных заемщиков составляет 282686 к 24825 соответственно. Доля дефолтных заемщиков 8%. Так как объем обучающей выборки достаточно большой, как и количество объектов миноритарного класса в абсолютном выражении, то для вышеупомянутых алгоритмов для решения проблемы соотношения классов будет использовано случайное удаление объектов мажоритарного класса. То есть объем обучающей выборки будет $24825 \times 2 = 49650$.

Для моделей случайного леса, градиентного бустинг и наивного байесовского классификатора данная проблема не является существенной.

3.2. Результаты построенных моделей и их сопоставительный анализ

В данном параграфе для каждого построенного алгоритма будут приведены характеристики их качества, такие как графики PR-кривой и ROC-кривой, площадь под ROC-кривой, точность, полнота, F-мера при стандартном предсказании модели, а также подобран порог, максимизирующий F-меру для первого класса (дефолт), и эти же метрики при данном пороге [35]. Все

вышеперечисленные результаты посчитаны на одном и том же наборе данных, на котором не обучалась ни одна модель, то есть на отложенной выборке.

Логистическая регрессия. Полученная модель выглядит следующим образом:

$$\begin{gathered} \boldsymbol{f(x) = \frac{1}{1 + e^{-z}},} \\ \boldsymbol{z = 0.57 - 0.25x_1 + 0.0006x_2 - 0.85x_3 - 0.92x_4} \\ \boldsymbol{- 2.71x_5 - 0.52x_6 - 0.55x_7 + 0.48x_8} \\ \boldsymbol{-0.37x_9 - 0.003x_{10} + 0.003x_{11} + 2.60x_{12}} \\ \boldsymbol{+0.21x_{13} - 0.08x_{14},} \end{gathered} \tag{37}$$

где:

x_1 – количество детей;

x_2 – доход клиента в долларах в год;

x_3 – сумма по кредиту;

x_4 – возраст клиента;

x_5 – за сколько дней до подачи заявления клиент начал текущую работу;

x_6 – за сколько дней до подачи заявления клиент изменил свою регистрацию;

x_7 – за сколько дней до подачи заявления клиент изменил документ, удостоверяющий личность, с которым он подал заявку на кредит;

x_8 – пол;

x_9 – наличие собственного автомобиля;

x_{10} – наличие собственного дома или квартиры;

x_{11} – предоставил ли клиент мобильный телефон;

x_{12} – предоставил ли клиент рабочий телефон;

x_{13} – был ли мобильный телефон доступен;

x_{14} – предоставил ли клиент электронную почту.

Из формулы модели выше следует, что данной линейная комбинация z аппроксимирует логарифм отношения вероятности того, что объект принадлежит классу $y = 1$ к вероятности того, что объект принадлежит классу

$y = 0$. Данное отношение называется *риском*, а логарифм от риска называется *логитом*. Следовательно, коэффициенты в модели можно интерпретировать как эластичность логита по признаку при данном коэффициенте.

Самыми значимыми признаками являются:

- за сколько дней до подачи заявления клиент начал текущую работу;
- предоставил ли клиент рабочий телефон;
- возраст клиента;
- сумма по кредиту;
- за сколько дней до подачи заявления клиент изменил документ, удостоверяющий личность, с которым он подал заявку на кредит.

Ниже, на рисунках 5 и 6, изображены кривые точности и полноты и ROC кривые соответственно:

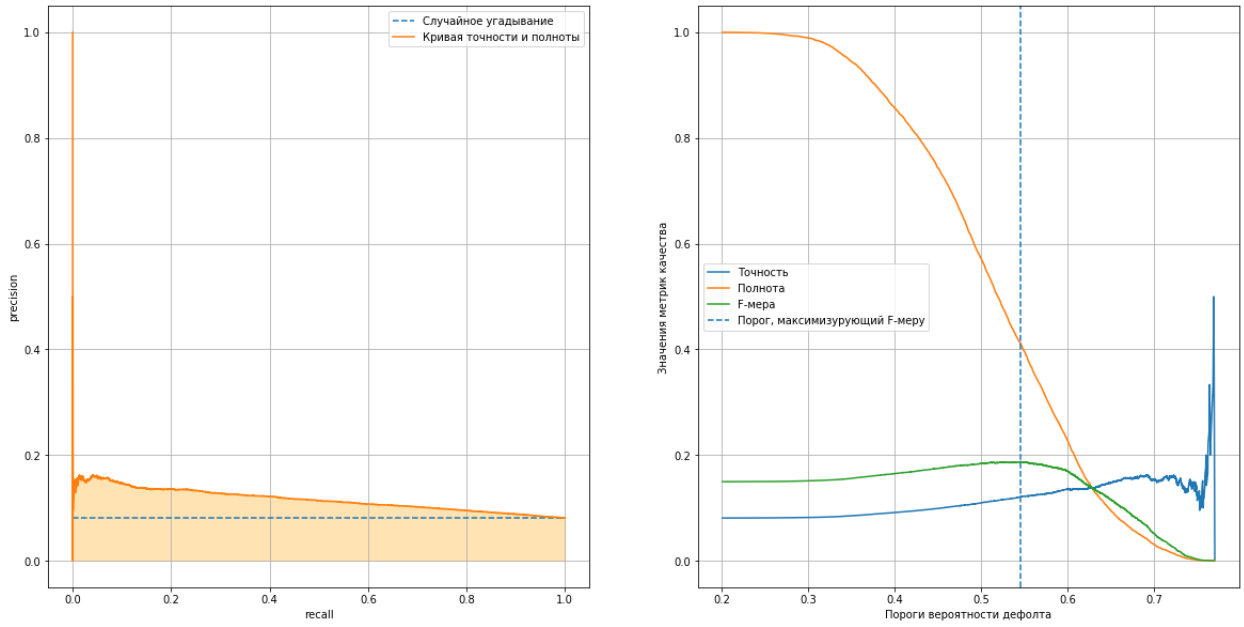

Рисунок 5 – Кривые точности и полноты для логистической регрессии

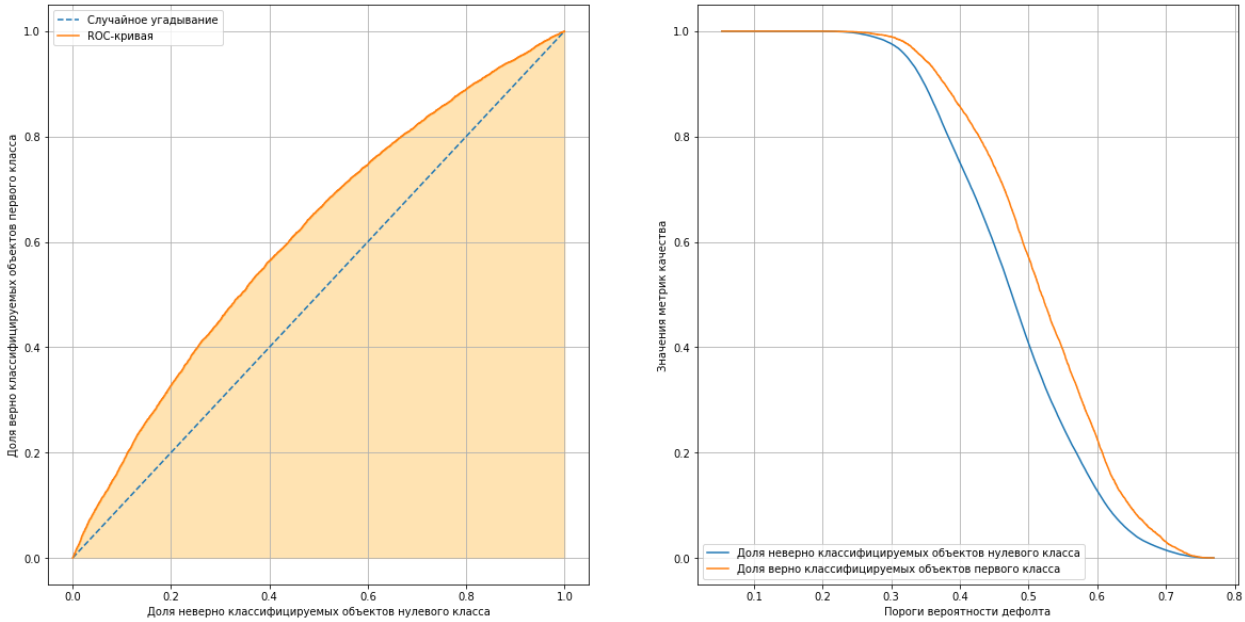

Рисунок 6 – ROC кривые для логистической регрессии

Метрики качества для данной модели показаны в таблице 7:

Таблица 7 – Метрики классификации для логистической регрессии без выбора порога

Класс / Метрика	**Точность**	**Полнота**	***F*-мера**	**ROC-AUC**	**Количество объектов**
0 – не дефолт	0.94	0.59	0.73	0.61	92878
1 – дефолт	0.11	0.57	0.18		8169

Максимум F-меры достигается при пороге 0.54 (см рис. 5). Значения метрик при данном пороге представлены в таблице 8:

Таблица 8 – Метрики классификации для логистической регрессии с выбором порога

Класс / Метрика	**Точность**	**Полнота**	***F*-мера**	**ROC-AUC**	**Количество объектов**
0 – не дефолт	0.93	0.74	0.82	0.61	92878
1 – дефолт	0.12	0.41	0.19		8169

Метод опорных векторов. Гиперпараметр регуляризации C = 1. Полученная модель выглядит следующим образом:

$$\begin{gathered} \boldsymbol{sgn}(\boldsymbol{z}), \\ \boldsymbol{z = 1.33x_1 + 0.1x_2 - 1.34x_3 - 1.65x_4} \\ \boldsymbol{-6.22x_5 - 0.48x_6 - 0.99x_7 + 0.82x_8} \\ \boldsymbol{-0.61x_9 + 0.03x_{10} + 0x_{11} + 5.96x_{12}} \\ \boldsymbol{-0.30x_{13} - 0.21x_{14} - 1.57}, \end{gathered} \tag{38}$$

где:

x_1 – количество детей;

x_2 – доход клиента в долларах в год;

x_3 – сумма по кредиту;

x_4 – возраст клиента;

x_5 – за сколько дней до подачи заявления клиент начал текущую работу;

x_6 – за сколько дней до подачи заявления клиент изменил свою регистрацию;

x_7 – за сколько дней до подачи заявления клиент изменил документ, удостоверяющий личность, с которым он подал заявку на кредит;

x_8 – пол;

x_9 – наличие собственного автомобиля;

x_{10} – наличие собственного дома или квартиры;

x_{11} – предоставил ли клиент мобильный телефон;

x_{12} – предоставил ли клиент рабочий телефон;

x_{13} – был ли мобильный телефон доступен;

x_{14} – предоставил ли клиент электронную почту.

Самыми значимыми признаками являются:

- за сколько дней до подачи заявления клиент начал текущую работу;
- предоставил ли клиент рабочий телефон;
- возраст клиента;
- сумма по кредиту;

- количество детей.

Самые значимые признаки метода опорных векторов совпадают с самыми важными признаками модели логистической регрессии за исключением признака *количество детей*.

Ниже, на рисунках 7 и 8, изображены кривые точности и полноты и ROC кривые соответственно:

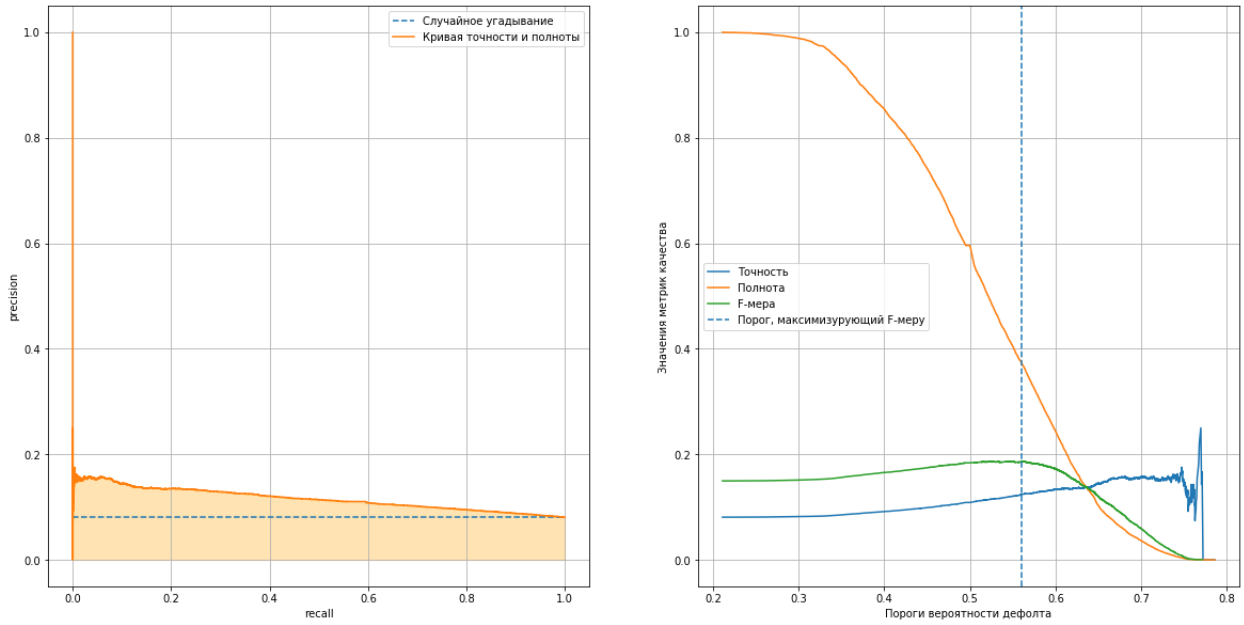

Рисунок 7 – Кривые точности и полноты для метода опорных векторов

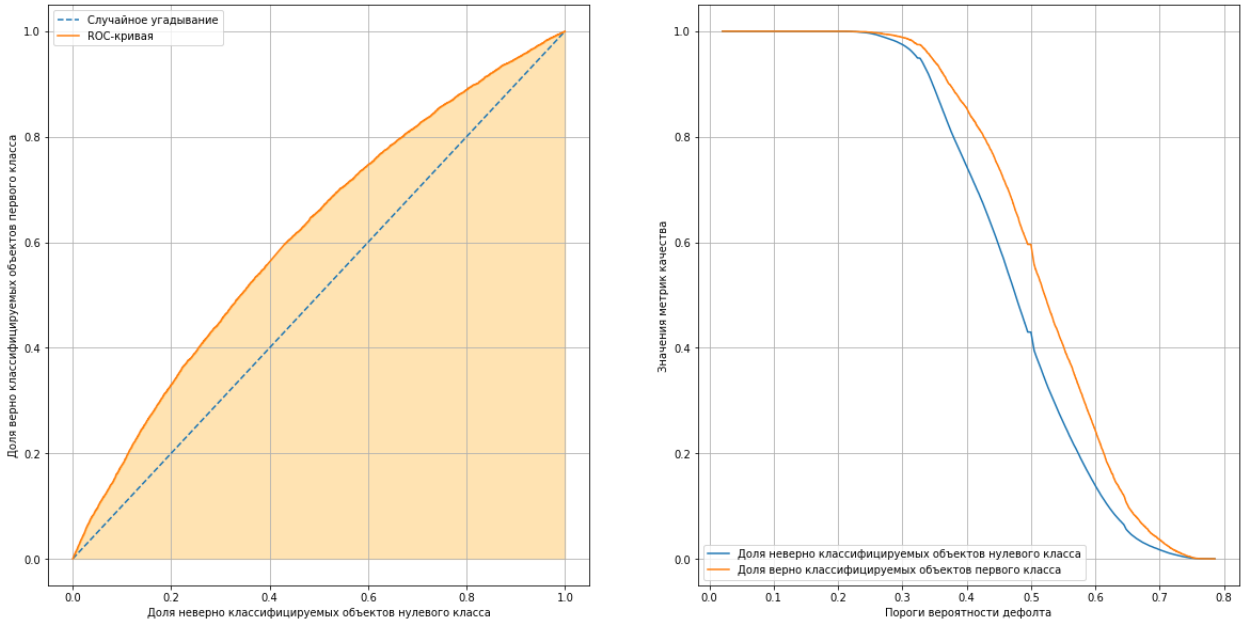

Рисунок 8 – ROC кривые для метода опорных векторов

Метрики качества для данной модели представлены в таблице 9:

Таблица 9 – Метрики классификации для метода опорных векторов без выбора порога

Класс / Метрика	**Точность**	**Полнота**	***F*-мера**	**ROC-AUC**	**Количество объектов**
0 – не дефолт	0.94	0.58	0.72	0.61	92878
1 – дефолт	0.11	0.59	0.18		8169

Максимум *F*-меры достигается при пороге 0.56 (см рис. 7). Метрики качества при данном пороге представлены в таблице 10:

Таблица 10 – Метрики классификации для метода опорных векторов с выбором порога

Класс / Метрика	**Точность**	**Полнота**	***F*-мера**	**ROC-AUC**	**Количество объектов**
0 – не дефолт	0.93	0.77	0.84	0.61	92878
1 – дефолт	0.12	0.38	0.19		8169

Метод k ближайших взвешенных соседей. Гиперпараметр количество соседей равен 20. Ниже изображены кривые точности и полноты и ROC кривые:

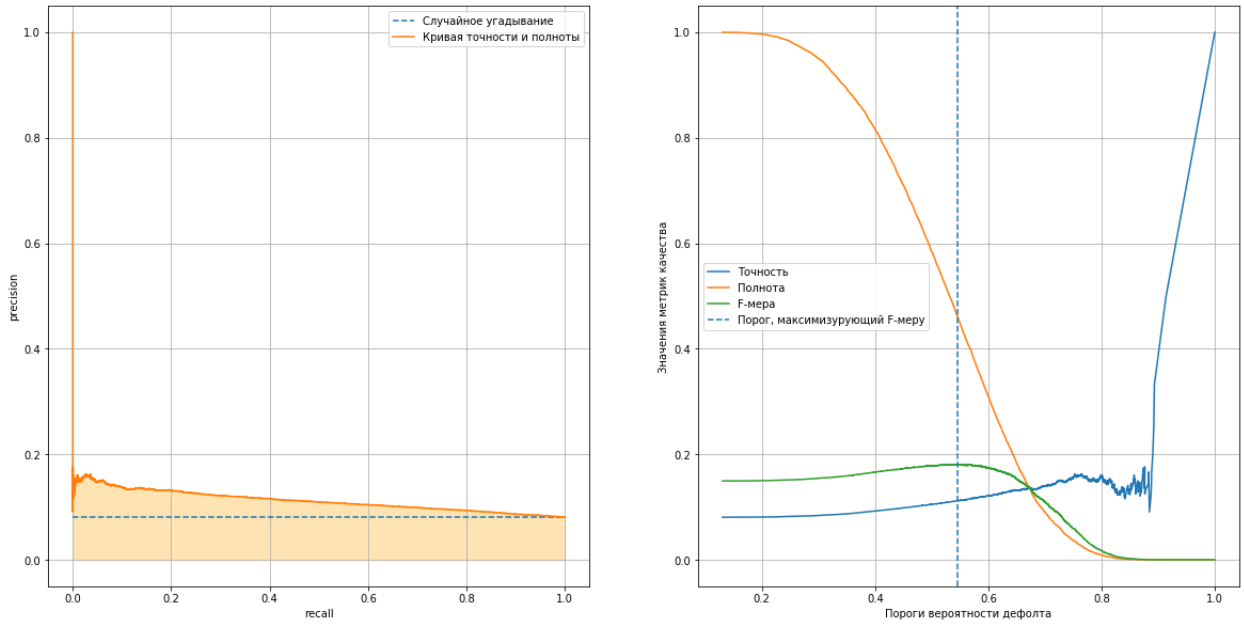

Рисунок 9 – Кривые точности и полноты для метода *k* ближайших взвешенных соседей

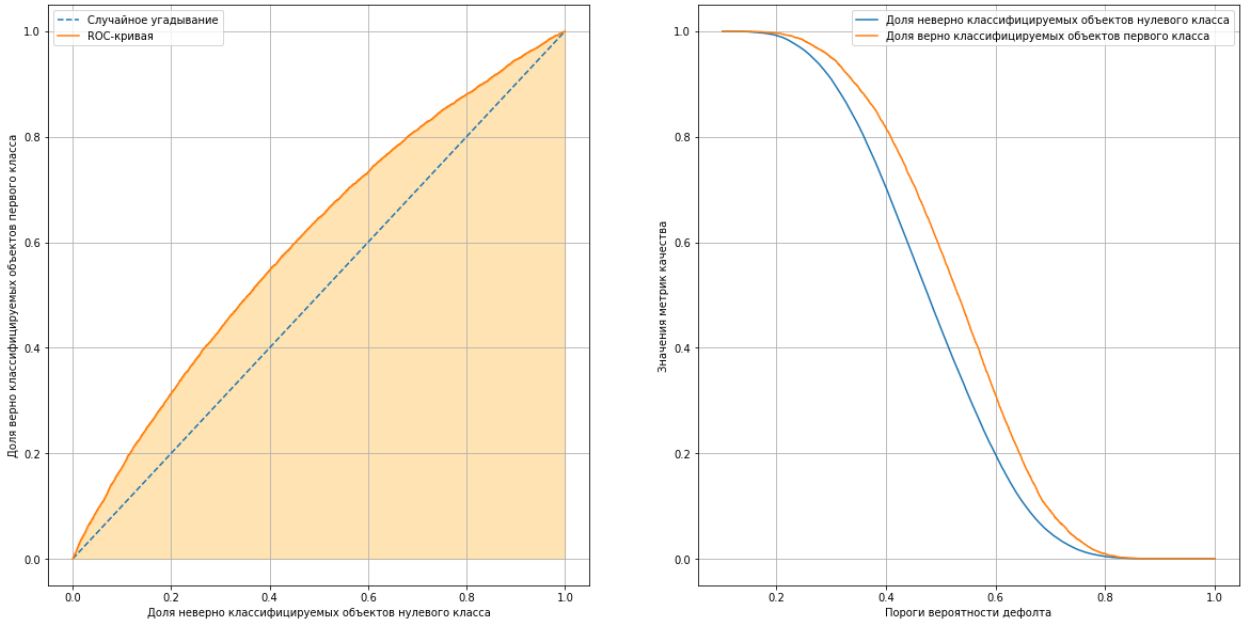

Рисунок 10 - ROC-кривые для метода *k* ближайших взвешенных соседей

В отличие от логистической регрессии и метода опорных векторов, данная модель не показывает значимости признаков и не дает интерпретации результатов.

Метрики качества для данной модели представлены в таблице 11:

Таблица 11 – Метрики классификации для метода *k* ближайших взвешенных соседей без выбора порога

Класс / Метрика	**Точность**	**Полнота**	***F*-мера**	**ROC-AUC**	**Количество объектов**
0 – не дефолт	0.94	0.56	0.70	0.60	92878
1 – дефолт	0.11	0.58	0.18		8169

Максимум *F*-меры достигается при пороге 0.54 (см рис. 9). Метрики качества при данном пороге представлены в таблице 12:

Таблица 12 – Метрики классификации для метода *k* ближайших взвешенных соседей с выбором порога

Класс / Метрика	**Точность**	**Полнота**	***F*-мера**	**ROC-AUC**	**Количество объектов**
0 – не дефолт	0.93	0.68	0.79	0.60	92878
1 – дефолт	0.11	0.46	0.18		8169

Наивный байесовский классификатор. Данная модель для каждого признака находит апостериорное распределения вероятности, но эту информацию нельзя использовать для определения значимости признаков и интерпретации работы модели. Ниже, на рисунках 11 и 12, изображены кривые точности и полноты и ROC кривые соответственно:

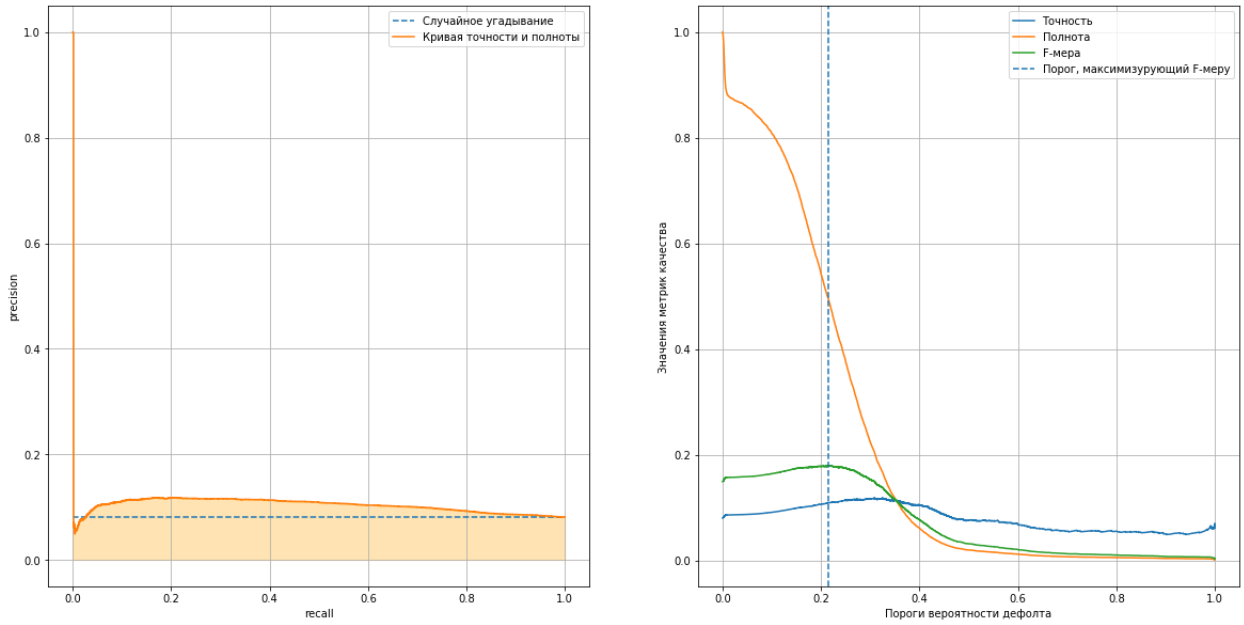

Рисунок 11 – Кривые точности и полноты для наивного байесовского классификатора

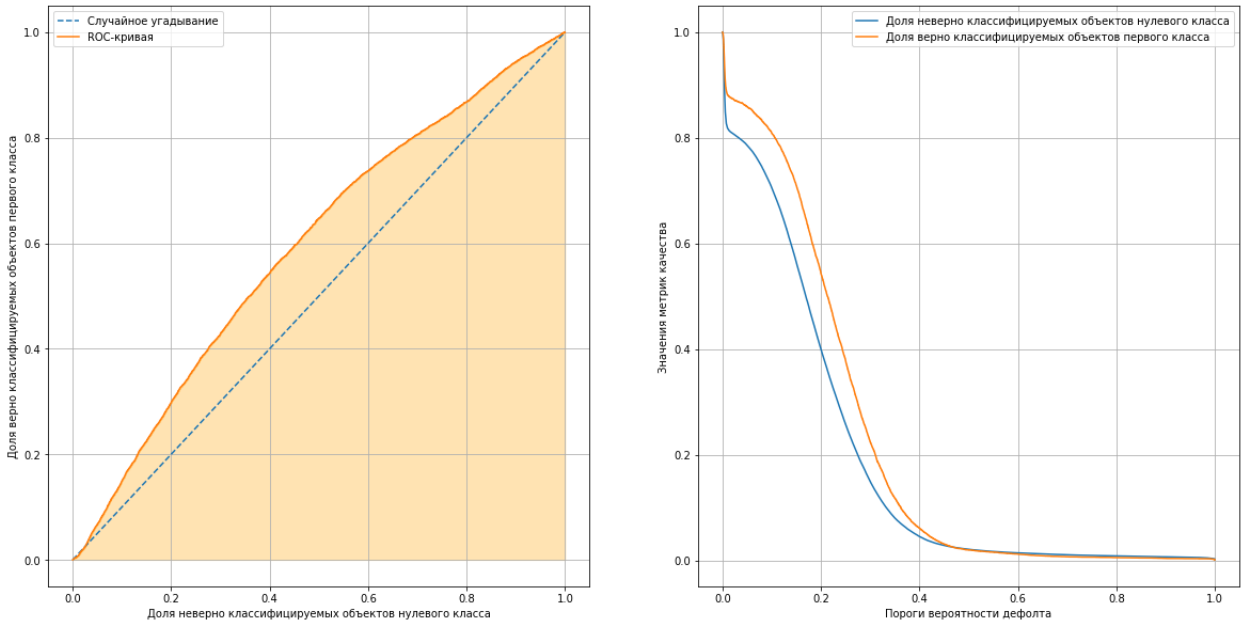

Рисунок 12 – ROC кривые для наивного байесовского классификатора

Метрики качества для данной модели представлены в таблице 13:

Таблица 13 – Метрики классификации для наивного байесовского классификатора без выбора порога

Класс / Метрика	**Точность**	**Полнота**	***F*-мера**	**ROC-AUC**	**Количество объектов**
0 – не дефолт	0.92	0.98	0.95	0.59	92878
1 – дефолт	0.08	0.02	0.03		8169

Максимум *F*-меры достигается при пороге 0.21 (см рис. 11). Метрики качества при данном пороге представлены в таблице 14:

Таблица 14 – Метрики классификации для наивного байесовского классификатора с выбором порога

Класс / Метрика	**Точность**	**Полнота**	***F*-мера**	**ROC-AUC**	**Количество объектов**
0 – не дефолт	0.94	0.65	0.76	0.59	92878
1 – дефолт	0.11	0.49	0.18		8169

Случайный лес. Алгоритм случайного леса обучен на 1000 базовых деревьях. Вид модели не приводится ввиду его сложности. Самыми значимыми признаками для данной модели являются:

- возраст;
- за сколько дней до подачи заявления клиент изменил документ, удостоверяющий личность, с которым он подал заявку на кредит;
- за сколько дней до подачи заявления клиент изменил свою регистрацию;
- сумма по кредиту;
- за сколько дней до подачи заявления клиент начал текущую работу.

Ниже, на рисунках 13 и 14, изображены кривые точности и полноты и ROC кривые соответственно:

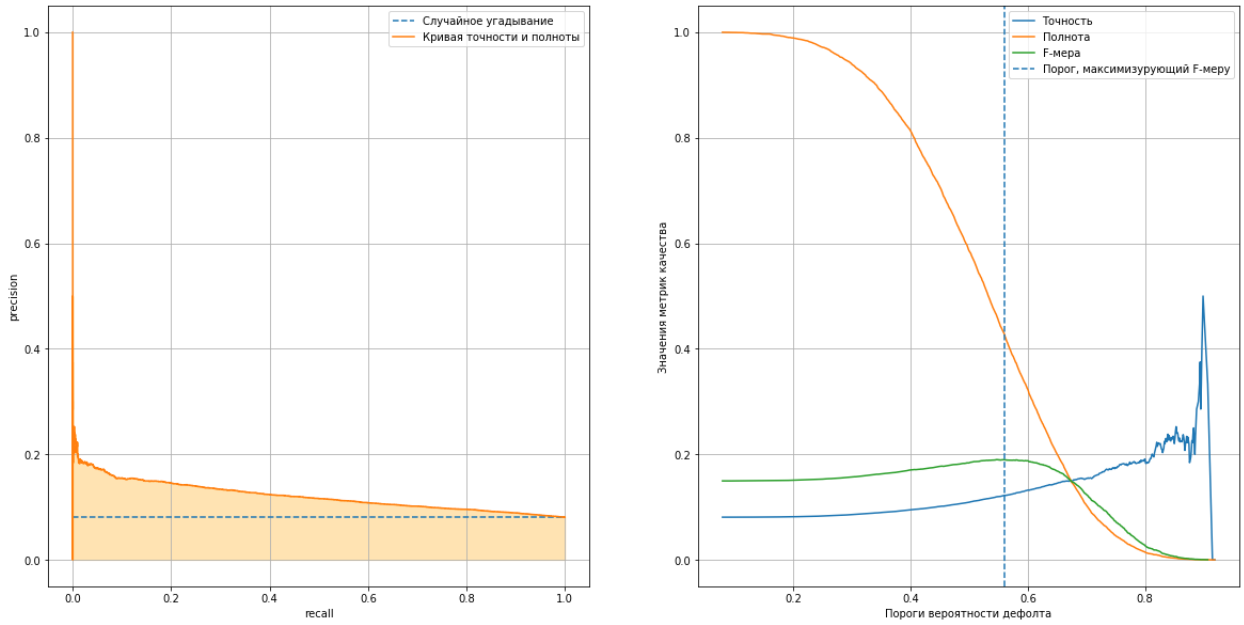

Рисунок 13 – Кривые точности и полноты для случайного леса

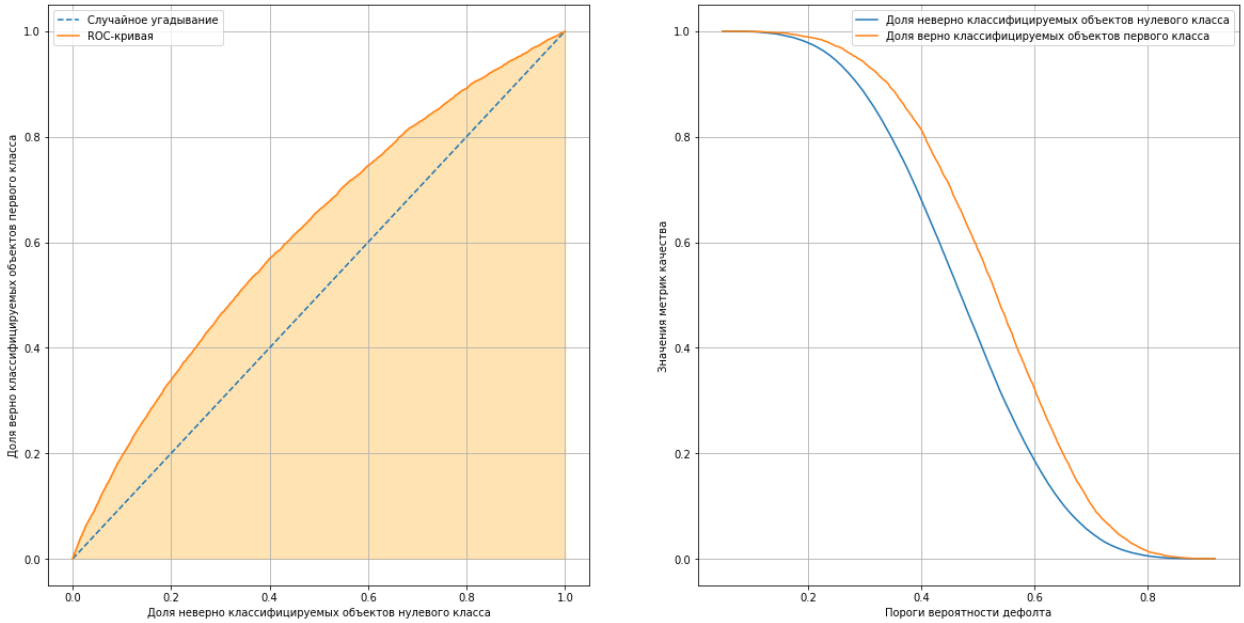

Рисунок 14 – ROC кривые для случайного леса

Метрики качества для данной модели представлены в таблице 15:

Таблица 15 – Метрики классификации для случайного леса без выбора порога

Класс / Метрика	**Точность**	**Полнота**	***F*-мера**	**ROC-AUC**	**Количество объектов**
0 – не дефолт	0.94	0.58	0.72	0.61	92878
1 – дефолт	0.11	0.58	0.18		8169

Максимум F-меры достигается при пороге 0.56 (см рис. 13). Метрики качества при данном пороге представлены в таблице 16:

Таблица 16 – Метрики классификации для случайного леса с выбором порога

Класс / Метрика	**Точность**	**Полнота**	***F*-мера**	**ROC-AUC**	**Количество объектов**
0 – не дефолт	0.94	0.65	0.76	0.61	92878
1 – дефолт	0.11	0.49	0.18		8169

Градиентный бустинг. Вид модели, также, как и для случайного леса, не приводится ввиду сложности. Лучшие гиперпараметры:

количество деревьев (базовых алгоритмов) n_estimators: 600;

коэффициент скорости обучения learning_rate: 0.02;

соотношение подвыборок столбцов при построении каждого дерева colsample_bytree: 0.8;

минимальное уменьшение потерь, необходимое для создания дальнейшего разбиения на листовом узле дерева gamma: 1.5;

максимальная глубина дерева для базовых алгоритмов max_depth: 4;

минимальная сумма веса экземпляра (гессиана), необходимая для листа min_child_weight: 10;

коэффициент подвыборки для тренировки subsample: 0.6.

Самыми значимыми признаками для данной модели являются:

- пол;
- наличие собственного автомобиля;
- возраст;
- за сколько дней до подачи заявления клиент начал текущую работу;
- за сколько дней до подачи заявления клиент изменил документ, удостоверяющий личность, с которым он подал заявку на кредит.

Модель градиентного бустинга единственная, у которой в одним из самых значимых факторов является *пол*.

Ниже, на рисунках 15 и 16, изображены кривые точности и полноты и ROC кривые соответственно:

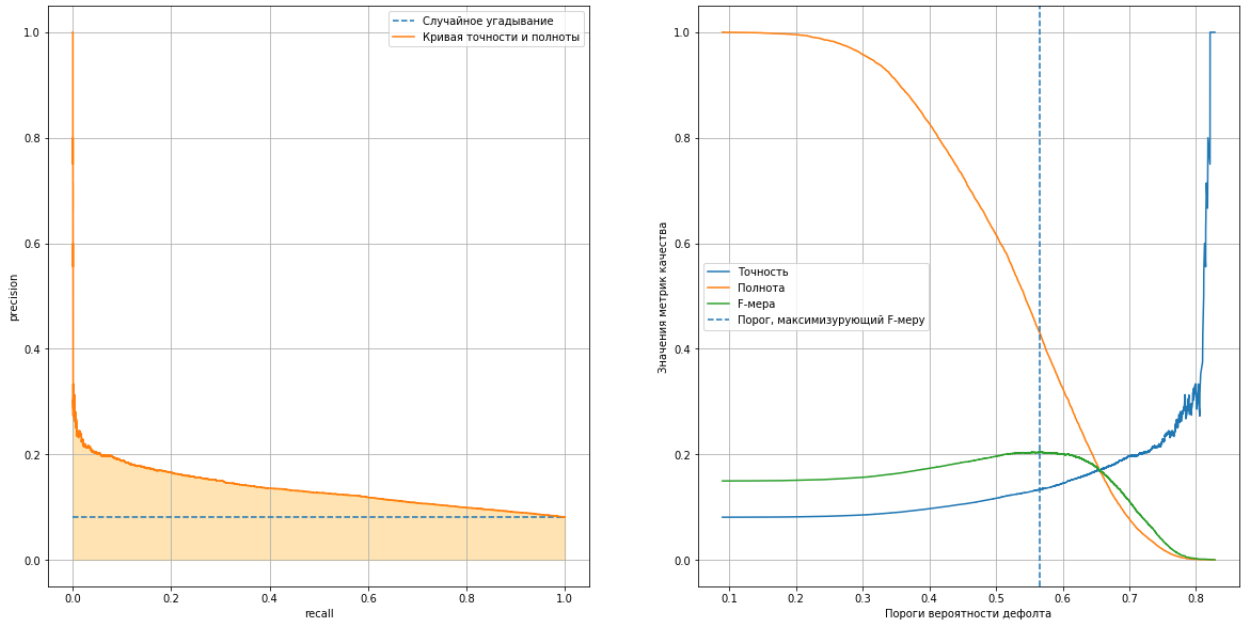

Рисунок 15 – Кривые точности и полноты для градиентного бустинга

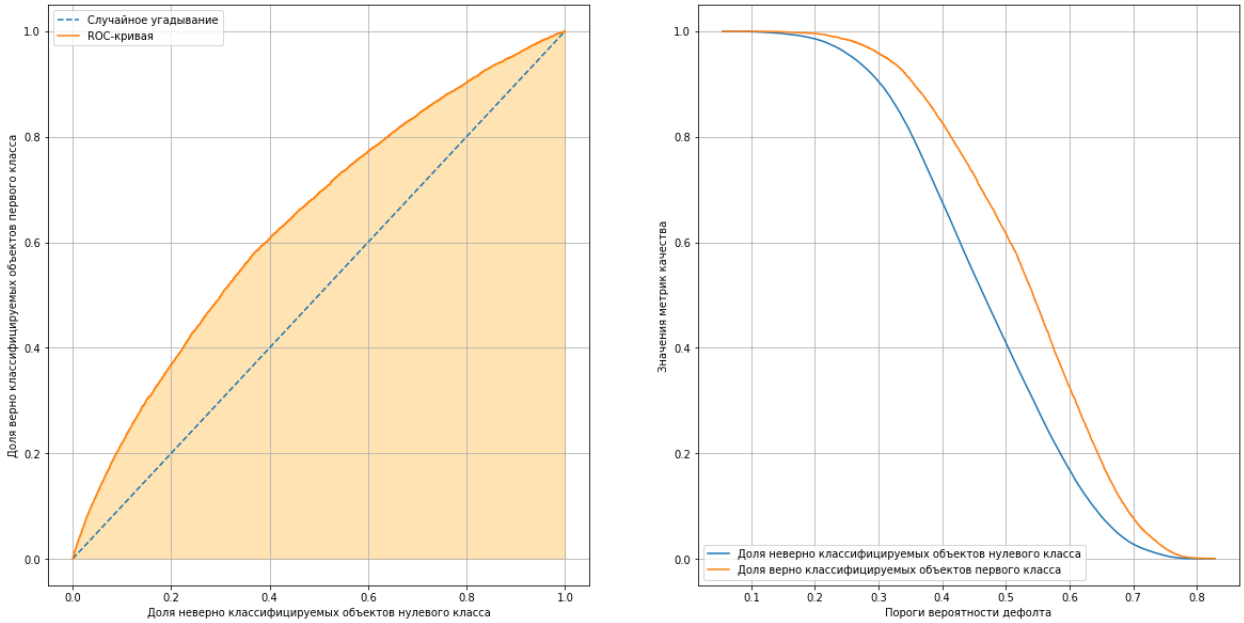

Рисунок 16 – ROC кривые для градиентного бустинга

Метрики качества для данной модели представлены в таблице 17:

Таблица 17 – Метрики классификации для градиентного бустинга без выбора порога

Класс / Метрика	**Точность**	**Полнота**	***F*-мера**	**ROC-AUC**	**Количество объектов**
0 – не дефолт	0.95	0.59	0.73	0.64	92878
1 – дефолт	0.12	0.62	0.20		8169

Матрица ошибок для данной модели представлены в таблице 18:

Таблица 18 – Матрица ошибок для градиентного бустинга без выбора порога

	$\boldsymbol{y = 1}$	$\boldsymbol{y = 0}$
$\boldsymbol{a(x) = 1}$	5035	38057
$\boldsymbol{a(x) = 0}$	3134	54821

Максимум F-меры достигается при пороге 0.56 (см рис. 15). Метрики качества при данном пороге представлены в таблице 19:

Таблица 19 – Метрики классификации для градиентного бустинга с выбором порога

Класс / Метрика	**Точность**	**Полнота**	***F*-мера**	**ROC-AUC**	**Количество объектов**
0 – не дефолт	0.94	0.76	0.84	0.64	92878
1 - дефолт	0.13	0.43	0.21		8169

Матрица ошибок для модели градиентного бустинга при пороге 0.56 представлены в таблице 20:

Таблица 20 – Матрица ошибок для градиентного бустинга с выбором порога

	$\boldsymbol{y = 1}$	$\boldsymbol{y = 0}$
$\boldsymbol{a(x) = 1}$	3509	22641
$\boldsymbol{a(x) = 0}$	4660	70237

Ниже представлены результаты по всем моделям:

Таблица 21 – Сопоставительная таблица метрик всех алгоритмов

Алгоритм / Метрика	**ROC-AUC**	**максимум *F*-меры**
Логистическая регрессия	0.61	0.19
Метод опорных векторов	0.61	0.19
Метод *k* ближайших взвешенных соседей	0.60	0.18
Наивный байесовский классификатор	0.59	0.18
Случайный лес	0.61	0.18
Градиентный бустинг	***0.64***	***0.21***

Из таблицы видно, что лучшее качество дает модель градиентного бустинга, поэтому именно она будет использоваться для прогнозирования дефолтности новых заемщиков. Для первых 5 заемщиков обучающей выборки, которые были представлены ранее, получаются следующие оценки:

Таблица 22 – Результат работы модели на первых 5 заемщиках обучающей выборки

Номер клиента	Факт дефолта	Вероятность дефолтности по модели градиентного бустинга
100002	1	0.73
100003	0	0.43
100004	0	0.54
100006	0	0.36
100007	0	0.57

Видно, что дефолтный заемщик получил оценку дефолтности больше, чем другие. Рассмотрим результаты работы модели на всех клиентах:

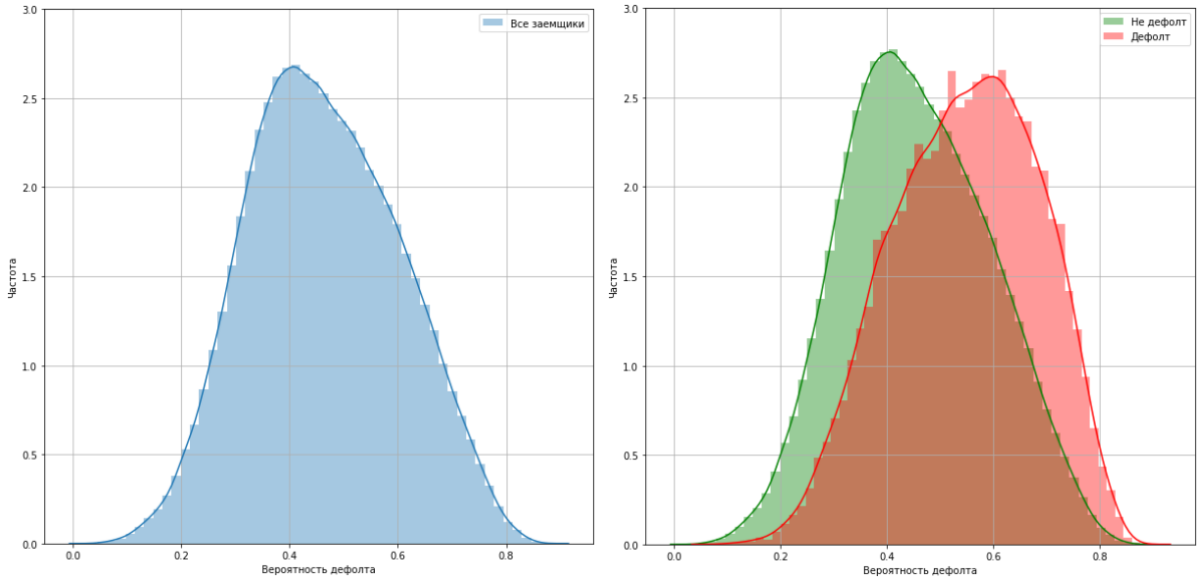

Рисунок 17 – Распределения вероятностей дефолта для всех заемщиков и для каждого класса по отдельности

Действительно, дефолтные заемщики в среднем получают вероятность дефолтности больше, чем кредитоспособные заемщики. Проверим гипотезу о равенстве средних вероятностей дефолтности для дефолтных и не дефолтных заемщиков:

$$\begin{cases} \boldsymbol{H_0: \mu_1 = \mu_2} \\ \boldsymbol{H_1: \mu_1 \neq \mu_2} \\ \boldsymbol{Z\left(X_1^{n_1}, X_2^{n_2}\right) = \dfrac{\bar{X}_1 - \bar{X}_2}{\sqrt{\dfrac{S_1^2}{n_1} + \dfrac{S_2^2}{n_2}}} \approx \sim St(v)}, \text{где} \\ \boldsymbol{v = \dfrac{\left(\dfrac{S_1^2}{n_1} + \dfrac{S_2^2}{n_2}\right)^2}{\dfrac{S_1^4}{n_1^2(n_1 - 1)} + \dfrac{S_2^4}{n_2^2(n_2 - 1)}}} \end{cases} \quad (39)$$

Так как $Z = 102.68 > St(v) = 1.64$, то гипотеза о равенстве средних вероятностей дефолтности для дефолтных и не дефолтных заемщиков отклоняется.

Самыми значимыми признаками для итоговой модели градиентного бустинга являются:

- пол;
- наличие собственного автомобиля;
- возраст;
- за сколько дней до подачи заявления клиент начал текущую работу;
- за сколько дней до подачи заявления клиент изменил документ. удостоверяющий личность, с которым он подал заявку на кредит.

Заключение

В рамках выполнения данной квалификационной работы был получен алгоритм прогнозирования кредитоспособности заемщиков кредитных организаций. В ходе разработки модели было выполнено следующее.

1) Изучены основные понятия, связанные с кредитным скорингом, такие, как кредитные организации, дефолтность, кредитоспособность. Рассмотрено понятие кредитного скоринга, его назначение и задачи, а также факторы кредитного скоринга.

2) Рассмотрены различные подходы и алгоритмы для построения скоринговой модели, такие, как линейные модели, в частности логистическая регрессия и метод опорных векторов, метрические алгоритмы и его честный случай метод k ближайшие взвешенных соседей и наивный байесовский классификатор. Изучен класс алгоритмов машинного обучения под названием решающие деревья и способы построения композиций на основе этого алгоритма.

3) Изучены основные подходы к оценке качества моделей классификации с помощью таких метрик качества, как точность, полнота, F-мера, площадь под кривой точности и полноты и площадь под ROC кривой.

4) Рассмотрены такие проблемы при работе с реальными данными, как пропуски в данных, предобработка количественный и качественных признаков для разных алгоритмов.

5) Проведен сопоставительный анализ всех методов и их результатов на основе метрик качества и изучены основные факторы, влияющие на кредитоспособность заемщика.

6) Выбран лучший алгоритм на основе площади под ROC кривой и на основе этого алгоритма получены оценки вероятности дефолта для новых клиентов.

В результате получены модели, описывающие дефолтность заемщиков от различных факторов и позволяющие прогнозировать данное явление для новых клиентов.

Список используемых источников

1. Федеральный закон "О банках и банковской деятельности" от 02.12.1990 N 395-1

2. Положение о порядке формирования кредитными организациями резервов на возможные потери по ссудам, ссудной и приравненной к ней задолженности" (утв. Банком России 28.06.2017 N 590-П) (ред. от 26.12.2018)

3. Айвазян С.А., Бухштабер В.М., Енюков И.С., Мешалкин Л.Д. Прикладная статистика: Классификация и снижение размерности. М.: Финансы и статистика, 1989. 607 с.

4. Айвазян С.А., Енюков И.С., Мешалкин Л.Д. Прикладная статистика. Основы моделирования и первичная обработка данных. М.: Финансы и статистика, 1983. 471 с.

5. Андреева Г.В. Скоринг как метод оценки кредитного риска // Банковские технологии. 2000. № 6. с. 14–19.

6. Битков В.П. Основы банковского дела. Часть 1. Учебное пособие. - М.: МГИМО - Университет МИД России, 2005. - 104с.

7. Вапник В. Н. Восстановление зависимостей по эмпирическим данным. М.: Наука, 1979. 448 с.

8. Васильев Н.П., Егоров А.А. Опыт расчета параметров логистической регрессии методом Ньютона – Рафсона для оценки зимостойкости растений // Математическая биология и биоинформатика. 2011. т. 6. № 2. с. 190–199.

9. Воронцов К.В. Лекции по логическим алгоритмам классификации [Электронный ресурс]. – Ресурс доступа: http://www.machinelearning.ru/wiki/images/3/3e/Voron-ML-Logic.pdf (дата обращения: 25.05.2019).

10. Воронцов К.В. Лекции по методам оценивания и выбора моделей [Электронный ресурс]. – Ресурс доступа: http://www.machinelearning.ru/wiki/images/2/2d/Voron-ML-Modeling.pdf (дата обращения: 20.05.2019).

11. Воронцов К.В. Метрические алгоритмы классификации [Электронный ресурс]. – Ресурс доступа: http://machinelearning.ru/wiki/images/8/8f/Voron-ML-Metric1.pdf (дата обращения: 27.05.2019).

12. Вьюгин В.В. Математические основы теории машинного обучения и прогнозирования. М.: 2013. - 387 с.

13. Вьюгин В.В. Элементы математической теории машинного обучения: учеб. пособие. – М.: МФТИ: ИППИ РАН, 2010. – 231с.

14. Готовкин И. Комплексная Скоринговая модель оценки дефолта клиента // Банковские технологии. 2006. № 1. с. 27–35.

15. Дубров А.М., Мхитарян В.С., Трошин Л.И. Многомерные статистические методы: учеб. М.: Финансы и статистика, 2003. 352 с.

16. Елисеева И.И. Эконометрика: Учебник / Елисеева И.И., Курышева С.В., Костеева Т.В. и др. – М.: Финансы и статистика, 2007 – 575 с.

17. Иванова В.М. Байесовский подход в статистике // SWorld [Электронный ресурс]. – Ресурс доступа: https://www.sworld.com.ua/konfer37/756.pdf (дата обращения: 14.04.2019).

18. Ишина И.В., Сазонова М.Н. Скоринг – модель оценки кредитного риска // аудит и финансовый анализ. 2007. № 4. с. 297–304.

19. Литвинова С.А. Скоринговые системы как средство минимизации кредитного риска банка // аудит и финансовый анализ. 2010. № 2. с. 396–397.

20. Литтл Р.Дж.А., Рубин Д.Б. статистический анализ данных с пропусками. М.:Финансы и статистика, 1990. 336 с.

21. МФТИ курс «Обучение на размеченных данных» [Электронный ресурс]. – Ресурс доступа: https://www.coursera.org/learn/supervised-learning (дата обращения: 19.04.2019).

22. МФТИ курс «Поиск структуры в данных» [Электронный ресурс]. – Ресурс доступа: https://www.coursera.org/learn/unsupervised-learning (дата обращения: 20.05.2019).

23. НИУ ВШЭ курс «Введение в машинное обучение» [Электронный ресурс]. – Ресурс доступа: https://www.coursera.org/learn/vvedenie-mashinnoe-obuchenie (дата обращения: 08.04.2019).

24. Сорокин А.С. Построение скоринговых карт с использованием модели логистической регрессии // Науковедение [Электронный ресурс]. – Ресурс доступа: http://naukovedenie.ru/PDF/180EVN214.pdf (дата обращения: 03.05.2019).

25. Тихомиров Н.П., Тихомирова Т.М., Ушмаев О.С. Методы эконометрики и многомерного статистического анализа. – М.: Экономика, 2011. – 647 с.

26. Тихомирова Т.М. Модели дискретного выбора: учебное пособие / Т.М. Тихомирова, А.Г. Сукиасян. – М.: РУСАЙНС, 2018. – 208с

27. Якупов А.И. Применение деревьев решений для моделирования кредитоспособности клиентов коммерческого банка // Искусственный интеллект. 2008. № 4. с. 208–213.

28. Andreas C. Müller, Sarah Guido. Introduction to Machine Learning with Python: A Guide for Data Scientists // O'Reilly Media, Inc., 2016, 269 с.

29. Aurelien Geron. Hands-On Machine Learning with Scikit-Learn and TensorFlow // O'Reilly Media, Inc., 2017, 533 с.

30. Christine Bolton. Logistic regressions and its application in credit scoring. University of Pretoria, 2009, 240 с.

31. Georg Krempl. Adaptive Prediction Models and their Application to Credit Scoring. University of Graz, Austria, 2011, 163 с.

32. Home Credit Default Risk Dataset [Электронный ресурс]. – Ресурс доступа: https://www.kaggle.com/c/home-credit-default-risk (дата обращения: 12.03.2019).

33. HSE course «How to Win a Data Science Competition: Learn from Top Kagglers» [Электронный ресурс]. – Ресурс доступа: https://www.coursera.org/learn/competitive-data-science (дата обращения: 05.06.2019).

34. James Bergstra, Yoshua Bengio. Random Search for Hyper-Parameter Optimization // Journal of Machine Learning Research [Электронный ресурс]. – Ресурс доступа: http://jmlr.csail.mit.edu/papers/volume13/bergstra12a/bergstra12a.pdf (дата обращения: 02.02.2019).

35. Powers D.M.W. Evaluation: from Precision, Recall and F-measure to ROC, Informedness, Markednessand and Correlation // Journal of Machine Learning Technologies. 2011. Vol. 2. Iss. 1. P. 37–63.

36. Samuel Glasson. Censored Regression Techniques for Credit Scoring. RMIT University, 2007, 196 c.