МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра теории функций и стохастического анализа

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ОПТИМИЗАЦИИ КРЕДИТНОГО СКОРИНГА С ИСПОЛЬЗОВАНИЕМ ПРОЦЕДУРЫ БИННИНГА

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

Студентки 2 курса 248 группы
направления 09.04.03 — Прикладная информатика

механико-математического факультета
Радченко Екатерины Дмитриевны

Научный руководитель
доцент, к. ф.-м. н. _____ Н. Ю. Агафонова

Заведующий кафедрой
д. ф.-м. н., доцент _____ С. П. Сидоров

Саратов 2023

Введение. Кредитный скоринг представляет собой одно из наиболее успешных применений статистики в финансовой и банковской сфере. Его можно определить как набор моделей принятия решений и лежащих в их основе методов, которые определяют, кто получит кредит, в каком объеме и какие оперативные стратегии повысят прибыльность кредита для кредиторов [1]. Такой подход снижает стоимость и время обработки заявки, обеспечивает гибкость при решении компромисса между риском и увеличением продаж для финансового учреждения.

Процесс построения скоринговой карты состоит из нескольких этапов, среди которых поиск данных, их первичная обработка, преобразование непрерывных переменных, их отбор, построение модели в соответствии с избранным статистическим методом и, наконец, преобразование модели в систему показателей на основе баллов — скоринговую карту.

В качестве техники преобразования непрерывных переменных рассматривается биннинг. Биннинг — это процесс дискретизации путем извлечения небольших групп (бинов) из непрерывной переменной, каждой из которых назначается центральное значение, характеризующее интервал.

Цель работы — разработка программного обеспечения, позволяющего работать с кредитными данными, оптимизируя их при помощи различных процедур биннинга. **Актуальность** работы объясняется широкой распространенностью скоринговых карт в качестве средства оценки кредитоспособности банковских клиентов.

Для достижения поставленной цели необходимо решить следующие задачи:

- изучить теоретические основы построения скоринговых карт;
- изучить основные подходы к биннингу кредитных данных;
- разработать приложение для работы с кредитными данными, включая реализацию возможности применения различных техник биннинга, на языке Python;
- сравнить логистические модели, независимые переменные в которых дискретизированы при помощи различных алгоритмов биннинга.

Работа представлена следующей **структурой**.

В первой части работы исследуются ключевые моменты построения

скоринговых моделей: описывается постановка задачи, объясняются распространенные подходы к сбору, подготовке данных, построению и оценке качества моделей.

Вторая часть работы сфокусирована на одном из ключевых этапов построения скоринговых моделей — дискретизации данных. Биннинг кредитных данных раскрывается как неотъемлемая часть процесса создания скоринговой карты; исследуются распространненые в банковской сфере алгоритмы биннинга.

Третья часть работы раскрывает практические аспекты разработки приложения для работы с кредитными данными на языке Python. Особое внимание уделено разработке изученных алгоритмов биннинга, обеспечению возможности их применения на наборах данных.

В четвертой части сравниваются модели логистической регрессии, в которых для дискретизации непрерывных переменных используются различные алгоритмы биннинга.

Работа прошла апробацию на различных конференциях, в частности, на ежегодной студенческой конференции «Актуальные проблемы математики и механики», которую проводил механико-математический факультет СГУ в апреле 2023 года, в секции «Анализ данных», и в ноябре 2022 года на XI Международной молодежной научно-практической конференции «Математическое и компьютерное моделирование в экономике, страховании и управлении рисками».

Основное содержание работы. Раздел **«Кредитный скоринг: определение, назначение, процесс построения моделей»** обуславливает теоретическую базу скоринговых карт как конечной цели построения скоринговых моделей.

Предположим, что вектор $x = (x_1, x_2, \ldots, x_p)$ — вектор из p независимых переменных-предикторов, а y — целевая переменная. Переменные в векторе x могут быть непрерывными и категориальными. Переменная y бинарная, и принимает значение 0 в случае, если кредит выплачен; в противном случае $y = 1$. Главная задача кредитного скоринга — определить, является ли заемщик «плохим» или «хорошим», или предсказать вероятность того, что заемщик не выплатит кредит — $y = 1$ [2].

Чаще всего скоринговые модели строятся на основе логистической регрессии, которая является распространенной техникой для задачи бинарной классификации [3, 4].

Показатели эффективности моделей кредитного скоринга разнообразны, однако наиболее распространенной практикой является использование кривой ROC (Receiver Operating Characteristic) и площади под соответствующей кривой AUC (Area Under Curve) [5]. AUC измеряется от 0,5 до 1. Обычно считают, что значение площади от 0,9 до 1 соответствует отличному качеству модели, от 0,8–0,9 — очень хорошему, 0,7–0,8 — хорошему, 0,6–0,7 — среднему, 0,5–0,6 — неудовлетворительному [4].

Далее коэффициенты логистической регрессии используются для формирования баллов скоринговой карты, баллы суммируются для каждого клиента или счета, и итоговый балл вносится в скоринговую карту [4, 6].

В разделе **«Биннинг кредитных данных при построении скоринговых моделей»** объясняется ценность биннинга для построения скоринговых карт и описывается несколько распространенных алгоритмов биннинга.

Техника биннинга предполагает, что интервал взятых наблюдений разбивается на более мелкие интервалы — бины или группы, — и в качестве их характеристики каждому назначается центральное значение. Все наблюдения, лежащие в определенном субинтервале, формируют ассоциированный бин. Здесь и возникает задача, лежащая в основе биннинга — выбрать точки, определяющие центральную оценку (так называемый номер бина) [7].

При построеннии скоринговых карт биннинг необходим, поскольку созданные категории формируют кредитные баллы. Результаты биннинга должны рассматриваться как неотъемлемая часть окончательной модели [8, 9].

Исследован ряд часто используемых алгоримов биннинга.

Биннинг в равную ширину: наблюдаемые значения непрерывного признака сортируются, затем диапазон наблюдаемых значений переменной разделяется на k ячеек одинакового размера, где k — параметр, вводимый исследователем.

Биннинг в равный размер: атрибуты сначала сортируются, затем разбиваются на предопределенное количество бинов равного размера. Если значения x различны, все бины будут иметь одинаковое количество наблюдений, за

исключением последнего — в нём может быть меньше наблюдений. В случае, если в x есть повторяющиеся значения, эти значения должны быть вынесены в отдельный бин [10, 11].

Оптимальный биннинг состоит из трёх шагов:

1. Данные разбиваются на достаточно большое количество маленьких групп;
2. Для каждой соседствующей пары групп вычисляется p-значение;
3. Находится наибольшее p-значение для всех пар; если оно больше некоторого порога, пары объединяются, затем возврат к шагу 1; в противном случае алгоритм завершается [11].

Мультиинтервальный дискретизационный биннинг направлен на минимизацию энтропии. Функция энтропии основана на следующем. Пусть T — точка разбиения датасета S на S_1 и S_2. Пусть также даны k предикторов $C_1, \ldots, C_k$ и пусть $P(C_i, S)$ — доля записей в S, принадлежащих предиктору C_i. Функция энтропии принимает вид:

$$\mathrm{Ent}(S) = -\sum_{j=1}^{k} P(C_i, S) \log(P(C_i, S)). \tag{1}$$

При биннинге кредитных данных необходимо максимизировать энтропию по разбиениям S_1 и S_2:

$$\mathrm{Ent}(T, S) = \frac{|S_1|}{|S|}\mathrm{Ent}(S_1) + \frac{|S_2|}{|S|}\mathrm{Ent}(S_2). \tag{2}$$

Как только точка найдена для набора данных S, процесс повторяется для подинтервалов рекурсивно до тех пор, пока не будет достигнуто существенного улучшения в энтропии [11, 12].

Идея монотонного крупноячеистого классификатора с максимальным правдоподобием состоит в следующем: допустим, оценка вероятности дефолта уменьшается с ростом предиктора. Добиться этого можно, проведя следующую процедуру:

1. Взять наименьшее значение предиктора и добавлять к нему значение до тех пор, пока совокупная оценка вероятности дефолта не достигнет максимума. Это будет первая точка разбиения.
2. Начиная с этой точки подсчитать совокупную оценку вероятности де-

фолта, пока вновь не будет достигнут максимум. Это вторая точка разбиения.

3. Повторять 1 и 2, пока не будут получены все точки разбиения [1].

Биннинг Хи-объединение (Chi-Merge) схож с оптимальным биннингом и применяется к численным переменным. Разница в том, что для оценки схожести смежных бинов используется критерий χ^2:

$$\chi^2 = \sum_{i=1}^{m}\sum_{j=1}^{k}\frac{(A_{ij} - E_{ij})^2}{E_{ij}}, \tag{3}$$

где $m = 2$, k соответствуют количеству классов; A_{ij} — количество наблюдений в i-том интервале, j-том классе; $R_i = \sum\limits_{j=1}^{k} A_{ij}$ — количество наблюдений в i-том интервале; $C_j = \sum\limits_{i=1}^{k} A_{ij}$ — количество наблюдений в j-том классе; $N = \sum\limits_{j=1}^{k} C_j$ — общее число наблюдений; $E_{ij} = \frac{R_i C_j}{N}$ — ожидаемая частота A_ij [13, 14].

WoE (Weight of Evidence) — численный метод, комбинирующий доказательства в поддержку статистической гипотезы. Величина WoE формирует центральное значение (характеристику бина), заменяя им фактические значения независимых переменных [4]. Получить значение WoE можно в соответствии со следующей формулой:

$$\mathrm{WoE}_i = \ln\left(\frac{G_i}{\sum\limits_{i=1}^{n} G_i} \Big/ \frac{B_i}{\sum\limits_{i=1}^{n} B_i}\right), \tag{4}$$

где G_i — количество «хороших» наблюдений в бине i, B_i — «плохих» наблюдений в бине i [10].

Зная значение WoE, можно вычислить значение IV:

$$\mathrm{IV} = \sum_{i=1}^{n}\left[\mathrm{WoE}_i\left(\frac{G_i}{\sum\limits_{i=1}^{n} G_i} - \frac{B_i}{\sum\limits_{i=1}^{n} B_i}\right)\right]. \tag{5}$$

Если $\mathrm{IV} < 0,02$, предиктор не нужно использовать в модели (не обнаружено

значимого влияния на качество разделения двух групп клиентов); если $0,02 \leq \text{IV} < 0,1$, то связь между предиктором и целевой переменной слабая; если $0,1 \leq \text{IV} < 0,3$, то связь между предиктором и целевой переменной выражена средне; если $0,3 \leq \text{IV}$, предиктор и целевая переменная имеют сильную связь [15].

В разделе **«Разработка и тестирование приложения для работы с кредитными данными»** описан процесс взаимодействия с приложением для работы с кредитными данными на языке Python.

Создание приложения подразумевает описание модулей, реализующих непосредственно алгоритмы биннига кредитных данных, а также модулей, содержащих функции подсчета WoE, IV, статистических показателей, построения графиков и препроцессинга данных. Полученный продукт позволяет проводить биннинг кредитных данных в средах с поддержкой контейнеризации (Docker, OpenShift и т. д.).

При разработке приложения, помимо встроенных, активно использовались пакеты Pandas, NumPy,Matplotlib и Seaborn, Scikit-learn.

Приложение состоит из двух логических блоков: блока работы с данными и алгоритмами биннинга. Визуализация приложения в виде системы программных модулей представлена на рисунке 1 и 2. Для описания архитектуры использовался язык UML. В основе архитектуры приложения лежит объектно-ориентированная парадигма.

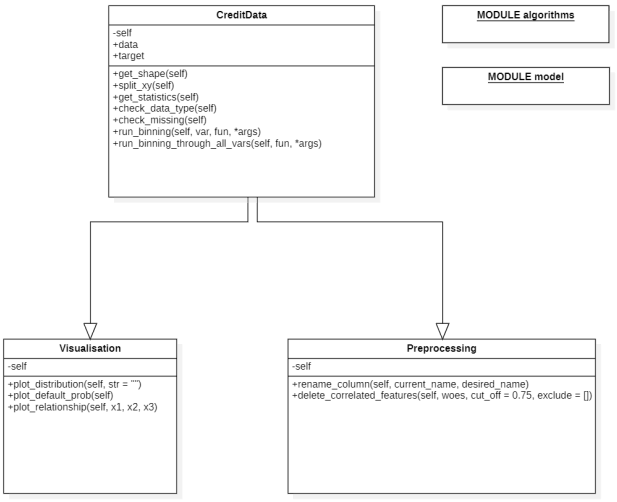

Рисунок 1 – Архитектура приложения: модуль работы с данными

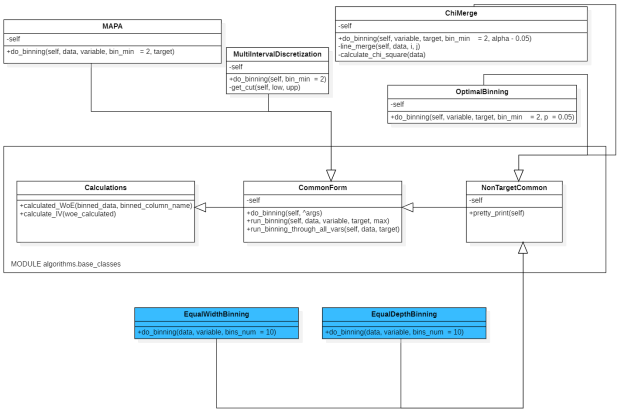

Рисунок 2 – Архитектура приложения: модуль algorithms

Модуль `base` содержит класс CreditData, включающий в себя методы первичной обработки и визуализации данных, реализованные в родительских классах `Preprocessing` и `Vizualization`. Они позволяют получать различную информацию о рассматриваемом наборе данных (описательные статистики, тип переменных, наличие пропущенных значений и т. д.), строить графики, важные для получения представления о данных, а также запускать процедуры биннинга относительно одной или множества переменных.

Конструктор класса CreditData содержит два поля: `data` — исходный набор данных, `target` — целевая переменная в строковом формате (название колонки в наборе данных).

Среди функций, реализованных в `base`, находятся функции запуска избранного алгоритма биннинга в двух вариантах: по всем переменным и по одной переменной. Чтобы запустить биннинг по одной переменной, необходимо вызвать функцию `run_binning()`. Она принимает два обязательных аргумента: функцию биннинга на первой позиции и переменную, для которой необходимо произвести операцию биннинга, на второй. Для неё также допустим ввод дополнительных аргументов. На третьей позиции — доля наблюдений, которая должна содержаться в бинах. Далее допустим ввод неограниченого количества аргументов, связанных с алгоритмом биннинга: например, значение Хи-квадрат или максимальное количество бинов, которые используются в функции, реализующей алгоритм объединения Хи-квадрат. Эти аргументы будут проброшены в соответствующий вызов функции биннинга.

Помимо непосредственной реализации возможности разбивать перемен-

ные на бины в соответствии с одним из распространенных алгоритмов, существует необходимость иметь инструмент для ручного регулирования границ первоначального автоматического разбиения на бины. Некоторая литература так же даёт рекомендацию предоставлять функционал для самостоятельного изменения разбиения [9]. Для ручной проработки границ так же полезно логирование промежуточных результатов автоматического биннинга. Для осуществления записи логов в функции биннинга добавлен параметр `log` со значением False по умолчанию: без указания `log = True` при вызове логи записываться не будут.

Родительский класс `Preprocessing` содержит функции переименования колонок и удаления коррелирующих значений. В классе `Vizualization`, также являющимся родительским по отношению к классу CreditData, реализованы методы визуализации данных: графики распределения, вероятности дефолта и взаимоотношения переменных.

Модуль `model` содержит функции `create_partition()`, осуществляющую разбиение данных на тестовую и тренировочную выборки в соответствии с методом отложенных данных, и `run_logit()`, строящую модель на основе логистической регрессии. Обе этих функции фактически являются обертками для функции `train_test_split()` из `sklearn.model_selection` и функций класса `LogisticRegression` из `sklearn.linear_model`.

Набор модулей `algorithms` содержит модули, описывающие основные алгоритмы биннинга. В него вложен набор `base_classes`, содержащий родительские классы для алгоритмов биннинга — них заданы интерфейсы для дальнейшего описания в потомках.

В качестве набора данных использовался `Give Me Some Credit`, содержащий исторические данные о 250 000 заемщиках (150 000 строк в файле `cs-training.csv` и 100 000 в `cs-test.csv`) [16].

При первичном тестировании приложения использовался алгоритм оптимального биннинга. AUC модели $\approx 0,84$, что соответствует очень хорошему качеству. Несколько строк скоринговой карты, полученной в результате вычислений на тестовом наборе данных `cs-test.csv`, представлены на рисунке 3.

	SeriousDlqin2yrs	BaseScore	x1	x2	x3	x4	x5	x6	x7	Score
46	0	430.0	-0.001	0.905	0.001	2.538	154.838	-0.004	-1.561	586.716
47	0	430.0	-0.001	0.156	-0.001	1.132	154.838	0.005	-1.561	584.568
48	1	430.0	-0.000	0.905	0.001	1.132	227.400	-0.001	-1.561	657.876
49	0	430.0	-0.000	-0.814	-0.001	0.538	154.838	-0.004	3.929	588.486
50	0	430.0	0.001	0.600	0.000	1.132	154.838	-0.004	0.084	586.651
51	0	430.0	0.001	0.425	0.000	0.538	154.838	-0.004	3.929	589.727

Рисунок 3 – Несколько строк .csv файла, хранящего данные скоринговой карты

Случайные 6 строк скоринговой карты демонстрируют результат вычисления скорингового балла и значение дефолта в течение двух лет для первых пяти клиентов из `cs-test.csv`. Ожидается, что кредиты 5 из 6 клиентов претерпят дефолт.

В разделе **«Сравнительный анализ моделей, полученных при использовании различных алгоритмов биннинга»** осуществляется сравнение логистических моделей, в основе которых использованы разные алгоритмы биннинга, по показателю AUC.

Параметры алгоритмов указаны в таблице 1.

Таблица 1 – Параметры алгоритмов биннинга

Алгоритм	**Параметры**
Биннинг в равную ширину	$k = 10$
Биннинг в равный размер	$k = 10$
Мультиинтервальный дискретизационный биннинг	—
Хи-объединение	$\alpha = 0.05$
Оптимальный биннинг	p-value = 0.05
Монотонный крупноячеистый классификатор	—

Для набора данных были рассмотрены шесть алгоритмов биннинга — оптимальный, в равную ширину и равный размер, монотонный крупноячеистый классификатор, мультиинтервальный и Хи-объединение. Было построено шесть моделей логистической регрессии, большинство из которых имели одинаковый состав переменных. Модели сравнивались по показателю AUC. Лучшей по этому критерию стала модель, независимые переменные в которой дискретизировались алгоритмом оптимального биннинга. Эта модель имеет имеет AUC = $0,85$, что соответствует очень хорошему качеству модели. Модели, оцененные как хорошие, использовали мультиинтервальный биннинг, биннинг в равную ширину и равный размер, достигая AUC в пределах $0,77$

и $0,78$. Модели среднего качества были получены при использовании алгоритмов Хи-объединение и MAPA. В их случае AUC не превышал $0,67$.

Заключение. Цель дипломной работы заключалась в разработке приложения для работы с кредитными данными, реализующим различные процедуры биннинга.

Были решены следующие задачи:

— изучены теоретические основы построения скоринговых карт;

— изучены основные подходы к биннингу кредитных данных;

— разработано приложение для работы с кредитными данными, включая реализацию возможности применения различных техник биннинга, на языке Python;

— проведено сравнение логистических моделей, независимые переменные в которых дискретизированы при помощи различных алгоритмов биннинга.

Приложение представлено двумя модулями: модулем работы с данными, включающим классы с методами первичной обработки данных, и обширным модулем с реализациями алгоритмов биннинга. Приложении позволяет провести процедуру создания скоринговой карты на основе логистической регрессии, управляя процессом дискретизации переменных. Приложение и шесть реализованных в нем алгоритмов биннинга были протестированы на наборе данных `Give Me Some Credit`. При сравнении моделей установлено, что наилучшей по AUC является модель, при дискретизации непрерывных переменных которой использовался алгоритм оптимального биннинга. Её AUC составляет $0,838$. Близкие к ней результаты были получены при использовании мультиинтервальной дискретизации — $AUC = 0,77$. Достаточно высокие показатели у простейших алгоритмов биннинга в ширину и равный размер: $AUC \approx 0,77$. AUC моделей при использовании алгоритмов Хи-объединение и MAPA не превысил $AUC \approx 0,67$.

Внедрение кредитного скоринга переориентирует цели компаний с минимизации потерь от каждого отдельного клиента на максимизацию общей прибыли. Разработанный продукт даёт возможность создания скоринговых карт, управляя процессом формирования баллов, что позволяет получать результаты для принятия решений о кредитовании.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 *Thomas, L.* Credit Scoring and Its Applications / L. Thomas, D. Edelman, J. Crook. — SIAM, 2002. — 262 pp.

2 *Abdou, H.* Credit scoring, statistical techniques and evaluation criteria: A review of the literature. / H. Abdou, J. Pointon // *Int. Syst. in Accounting, Finance and Management.* — 2011. — Vol. 18. — Pp. 59–88.

3 *Zheng, F.* A Vertical Federated Learning Method for Interpretable Scorecard and Its Application in Credit Scoring / F. Zheng, Erihe, K. Li, K. Tian, X. Xiang. — URL: `https://arxiv.org/abs/2009.06218`, / (Дата обращения 10.01.2022).— Загл. с экр.— Яз. англ. `https://arxiv.org/abs/2009.06218`.

4 *Сорокин, А. С.* Построение скоринговых карт с использованием модели логистической регрессии / А. С. Сорокин // *Труды Моск. матем. об-ва.* — 2014. — Т. 2.

5 *Kraus, A.* Recent Methods from Statistics and Machine Learning for Credit Scoring / A. Kraus. — Munich: Cuvillier, 2014. — 161 pp.

6 *Siddiqi, N.* Credit Risk Scorecards, Developing and Implementing Intelligent Credit Scoring / N. Siddiqi. — New Jersey: John Wiley and Sons, Inc., 2006. — 196 pp.

7 *Nisbet, R.* Handbook of Statistical Analysis and Data Mining Applications / R. Nisbet, G. Miner, K. Yalek. — London: Academic Press, 2017. — 822 pp.

8 *Verstraeten, G.* The impact of sample bias on consumer credit scoring performance and profitability / G. Verstraeten, D. V. den Poel // *Oper Res Soc.* — 2005. — Vol. 56.

9 *Szepannek, G.* An overview on the landscape of r packages for open source scorecard modelling / G. Szepannek // *Data Analysis for Risk Management – Economics, Finance and Business).* — 2022. — Vol. 10.

10 *Zeng, G.* A necessary condition for a good binning algorithm in credit scoring / G. Zeng // *Applied Mathematical Sciences.* — 2014. — Vol. 8, no. 65. — Pp. 227–246.

11 *Mironchyk, P.* Monotone optimal binning algorithm for credit risk modeling / P. Mironchyk, V. Tchistiakov. — URL: https://www.researchgate.net/publication/322520135_Monotone_optimal_binning_algorithm_for_credit_risk_modeling, / (Дата обращения 22.02.2022).— Загл. с экр.— Яз. англ. https://www.researchgate.net/publication/322520135_Monotone_optimal_binning_algorithm_for_credit_risk_modeling.

12 *Fayyad, U. M.* Multi interval discretization of continuous-valued attributes for classification learning / U. M. Fayyad, K. B. Irani // *Artificial intelligence.* — 1993. — Vol. 13, no. 65. — Pp. 1022–1027.

13 *Kerber, R.* Chimerge: Discretization of numeric attributes // In Proceedings of the Tenth National Conference on Artificial Intelligence. — Vol. 10. — 1992. — Pp. 123–128.

14 *Perner, P.* Data Mining on Multimedia Data / P. Perner. — Berlin: Springer Berlin Heidelberg, 2003. — 138 pp.

15 *Агафонова, Н. Ю.* Об одном методе оптимизации биннинга кредитных данных // Математическое моделирование и суперкомпьютерные технологии. Труды XX Международной конференции. под ред. проф. В.П. Гергеля. — Т. 10. — 2020. — С. 27.

16 UCI Machine Learning Repository: Default of credit card clients data set, [Электронный ресурс]. — URL: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients, / (Дата обращения 01.03.2022).— Загл. с экр.— Яз. англ. https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients.